SESSION

HIGH PERFORMANCE COMPUTING, DISTRIBUTED AND PARALLEL PROCESSING, DATABASES

Chair(s)

TBA

Benchmarking and Performance studies of MapReduce / Hadoop Framework on Blue Waters Supercomputer

Manisha Gajbe[1], **Kalyana Chadalavada**[1], **Gregory Bauer**[1], **William Kramer**[1,2]
[1]National Center for Supercomputing Applications, University Of Illinois at Urbana-Champaign, IL, USA
[2]University Of Illinois at Urbana-Champaign, IL, USA
$manisha, kalyan, gbauer, wtkramer$@illinois.edu

Abstract— MapReduce is an emerging and widely used programming model for large-scale data parallel applications that require to process large amount of raw data. There are several implementations of MapReduce framework, among which Apache Hadoop is the most commonly used and open source implementaion. These frameworks are rarely deployed on supercomputers as massive as Blue Waters. We want to evaluate how such massive HPC resource can help solving large-scale data analytics, data-mining problems using MapReduce / Hadoop framework.

In this paper we present our studies and detailed performance analysis of MapReduce / Hadoop framework on Blue Waters Supercomputer. We have used standard popular MapReduce benchmark suite that represents wide range of MapReduce applications with various computation and data densities. Also, we are planning to use Intel HiBench Haddop Benchamrk Suite in future. We identify few factors that significantly affect the performance of MapReduce / Hadoop and shed light on few alternatives that can improve the overall performance of MapReduce techniques on the system.

The results we have obtained strengthen our belief in possibility of using massive specialized superrcomputers to tackle big data problems. We demonstrate the initial performance of the MapReduce / Hadoop framework with encoraging results and we are confident that the massive traditional High Perfromance Computaing resource can be useful in tackling the big-data research challneges and in solving large-scale data analytics, data-mining problems.

Keywords: MapReduce, Hadoop, Blue Waters

1. Introduction

MapReduce [1] is a well-known programming framework pioneered by Google for data intensive and large-scale data analysis applications. The architecture is simple abstraction that allows programmers to use a functional programming style to create a *map* function that processes a key-value pair associated with the input data to generate a set of intermediate key-value pairs, and a *reduce* function that merges all intermediate values associated with the same intermediate key. The MapReduce programming model is divided into 3 simple phases namely: *Map; Shuffle and Sort; Reduce* as shown in figure 1.

Map Phase: In the map phase, the input data is partitioned into input splits and assigned to Map tasks associated with processing nodes in the cluster. The Map task typically executes on the same node containing its assigned partition of data in the cluster. These Map tasks perform user-specified computations on each input key-value pair from the partition of input data assigned to the task, and generate a set of intermediate results for each key.

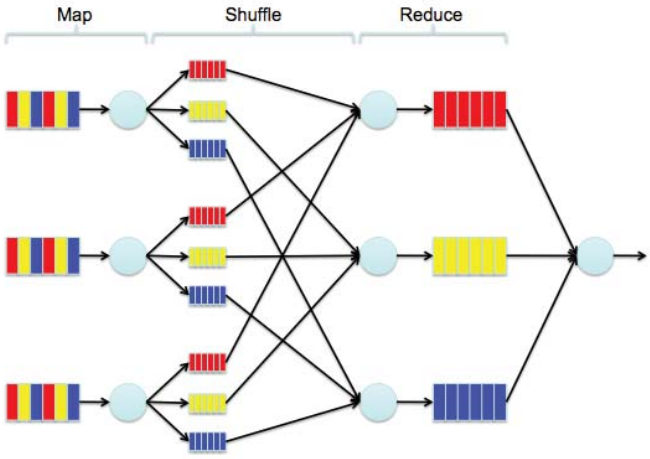


Fig. 1: MapReduce - 3 Phases

The shuffle and sort phase: The shuffle and sort phase sorts the intermediate data generated by each Map task from other nodes and divides this data into regions to be processed by the reduce tasks. This phase also distributes this data as needed to nodes where the Reduce tasks will execute.

Reduce Phase: In reduce phase, the data divided by shuffle and sort phase is processed. The Reduce tasks perform additional user-specified operations on the intermediate data possibly merging values associated with a key to a smaller set of values to produce the output data.

All Map tasks must complete prior to the shuffle and sort and reduce phases. The number of Reduce tasks does not need to be the same as the number of Map tasks. For more complex data processing procedures, multiple MapReduce calls may be linked together in sequence.

The MapReduce programming model is also becoming popular in scientific computing, where scientists need to frequently analyze a large volume of experimental and

simulation data. Such data analysis is often implemented as independent tasks that can be expressed as mapping operations in MapReduce. For example, in genome sequencing, the matching of large number of sequences against a huge collection of known sequences can be considered as mapping of similarity function to pair of sequences. Similarly, for the post-processing of simulation data, the tasks can be expressed as MapReduce, where a single program is run multiple times with different input parameters. MapReduce is a simple and scalable approach that enables scientists to achieve simulation results from large-scale data.

Google implemented MapReduce to execute very large matrix-vector multiplications needed for the PageRank calculations. Matrix operations such as matrix-vector and matrix-matrix calculations fit very well into the MapReduce style of computing. Another examples in numerical computing that can be solved are Singular Value Decomposition or Sparse Matrix Vector Multiplication that are used in lots of HPC applications. Given the large-scale problem size and types that are addressed using MapReduce, and the popularity of MapReduce as an implementation paradigm, it is unquestionable to explore its use on traditional HPC platforms.

In this paper, we conduct initial benchmarking and performance results of MapReduce framework on the Blue Waters [3] petascale system. We have briefly described the challenges of using Mapreduce / Hadoop framework on High Performance Computing (HPC) platforms. We have used Apache Hadoop [2], the most popular and commonly used MapReduce framework. However, there is no official / formal support for Hadoop or related stack on the Blue Waters system.

The rest of the paper is organized as follows: Section 2 discusses the challenges while deploying and using MapReduce / Hadoop on a High Perfoprmance Computing resource. In Section 3, we describe the architecture of computing systems used and brief description on the test cases and benchmarks we have experimented with. We talk about experimental setup along with benchmarking environment and Hadoop related configuration that we used and challenges faced in deploying Apache Hadoop software stack on the Blue Waters system in Section 4 follwed by results with discusion in Section 5 . Related work is briefly reviewed in Section 6 and finally we conclude and discuss future work in Section 7.

2. Challenges on HPC System

There are certain challenges on how MapReduce-Hadoop framework will fit into HPC environment. We have come across few of them while working with the MapReduce programing paradigm on the Blue Waters system.

2.1 Parallelism:

Most of the HPC applications use divide-conqure method and each task communicate with other tasks frequently. These applications are often classified according to how often their subtasks need to synchronize or communicate with each other. In applications that exhibit fine-grained parallelism the subtasks communicate frequently while applications with coarse-grained parallelism the subtasks do not communicate many times. Other types of HPC applications are embarrassingly parallel that rarely or never have to communicate. Embarrassingly parallel applications are considered the easiest to parallelize. MapReduce completely relies on Embarrassingly Parallel (EP) techniques. May of the HPC applications do not fall in this category. Programmers will need to rewrite the codes to expose the EP method in their existing codes. Also the programming models such as MPI, OpenMP, OpenACC etc developed for Parallel Programming are not suitable for MapReduce / Hadoop framework.

2.2 Programming Language:

The main programming language for HPC applications is either Fortran or C while Hadoop is written in Java so that the code written can run on any hardware platform. This is completely opposite when it comes to traditional HPC applications, where they are compiled and optimized for specific software and hardware platform on which they will run. As per the HPC users, the codes written in Java are slow and inefficient which is not acceptable in HPC coomunity. Another reason is, Hadoop was essentially designed for world wide web services, for which Java is almost perfect language of programming, while HPC applications address wide range of scientific applications that are developed historically.

2.3 File System:

The main requirement of Hadoop is availability of local data storage. However, for the HPC systems there is no local storage. The file system is shared across all the available nodes. Most of the time the file system is either General Parallel File System (GPFS) [32], [33] or Lustre [34]. Simulating these shared files system as a local storage is not straight forward. Additionally, these filesystems extensively use POSIX, while Hadoop doesn't support it.

2.4 Resource Management:

In traditional hadoop clusters the resource management is entirely handled by the hadoop framework and the types of workloads are similar. On the other hand, typical HPC systems handle various types of workload and the resource management is taken care by dedicated scheduling software such as Moab [35], PBS or Slurm. Integrating these resource scheduler with Hadoop is not a simple task.

2.5 Operating System:

Hadoop framework requires a full flavored operating system. Current HPC systems have stripped down version of linux kernel to reduce unwanted polling from the Operating System (OS) which in turn improves the performance of an application on the system. To use the MapReduce / Hadoop framework one will have to use the Cluster Compatibility Mode on the given system so that a full flavor of OS is available for MapReduce applications.

2.6 IP stack over interconnect:

Hadoop framework uses TCP / IP or Ethernet and not high speed and lossless Remote direct Memory Access (RDMA) technologies. Hadoop does not support low latency high speed interconnect with scalable topologies like 3D Torus or 5D torus or Dragonfly or Gemini etc. It supports only multi-stage clos style network [38]. The network topologies mentioned above are relevant only to HPC or Supercomputing.

We will have to look into all the above challenges to evaluate the MapReduce / Hadoop framework on Blue Waters system. However, solutions are being developed and attempted to address a few of the above mentioned challenges.

3. System Overview

In the section we describe the architecture of the computing system we have used for the benchmarking and performance evaluation of MapReduce framework and brief description of test cases and benchmarking candidates. We have used our Test and Development System as well as Blue Waters [3] system to perform our experiments.

3.1 Computing System Overview

The hardware we used is the sustained petascale system of Blue Waters [3] hosted at the University of IllinoisâĂŹs National Center for Supercomputing Applications (NCSA) and funded by NSF. Blue Waters is one of the largest computational resources in the world, serving NSF Science community researchers throughout the United States.

3.1.1 JYC Configuration

JYC is our Test and Development System (TDS) where we test and evaluate software and changes before we deploy it on Blue Waters. JYC is a single rack XE6m/XK6m. There are 96 total nodes with a aggregate peak compute performance of ~30.3 TF.

- 76 nodes are XE6 nodes with (1216 bulldozer modules, 2432 integer cores, 313 GF/node, 23.8 TF total):
  - two AMD Interlagos processors (16 Bulldozer modules total)
  - 64 GB of RAM
- 8 nodes are XK6 nodes with (64 bulldozer modules, 128 integer threads, 156 GF x86/node, 655 GF/Fermi, 6.5 TF total):
  - one AMD Interlagos processor (8 Bulldozer modules)
  - 32 GB of RAM
  - one NVIDIA Fermi GPU (with 6GB of RAM)
- The remaining 12 nodes are service nodes used for boot, sdb, LNET routers, login, and network gateway. All 96 nodes are on the gemini interconnect which is cabled as a 2D mesh 1D torus. The login and network gateways each have a dual-port 10Gb Ethernet NIC.
- JYC also has Lustre as underlying file system, Torque as resource manager and Cluster Compatibility mode.

3.1.2 Blue Waters Configuration

Blue Waters is a Cray XE6-XK7 supercomputing system managed by the National Center for Supercomputing Applications for the National Science Foundation. The system has a peak performance of 13.34 PF, aggregate IO throughput in excess of 1 TB/s, 26 PB online disk capacity and nearly 200 PB of nearline storage. Blue Waters contains two types of compute nodes: XE6 and XK7. There are 22,640 XE6 nodes and 4,224 XK7 nodes. Each XE6 node has two 16 core AMD 6276 CPUs, 64 GB of main memory. Each XK7 node has one 16 core AMD 6276 CPU, 32 GB of main memory and one Nvidia Kepler K20X graphics processing unit (GPU) with 6 GB of GDDR5 on-board memory. The compute and file system nodes are interconnected using the Cray Gemini high speed interconnection network. Two nodes share a single Gemini ASIC (Application-Specific Integrated Circuit), which contains two network interface controllers (NICs) and a YARC-2 router. The network is organized in a 24 X 24 X 24 3D torus topology The detailed architecture of the system is described in [42].

3.1.3 File System

The file system on Blue Waters will be built using Cray Sonexion 1600 Lustre appliances. The Cray Sonexion 1600 appliances provide the basic storage building block for the Blue Waters I/O architecture [41] and are referred to as a "Scalable Storage Unit" (SSU). Each SSU is RAID protected and is capable of providing up to 5.35 GB/s of IO performance and approximately 120TB of usable disk space. The *scratch* file system where the runs were made uses 180 (one hundred eighty) SSUs to provide 21.6 PB of usable disk storage and 963 GB/s IO performance. This file system can provide storage for up to 2 million file system objects. This file system is high performance, high capacity transient storage for applications.

3.1.4 Resource Management

The Blue Waters system uses TORQUE Resource Manager [36] integrated with the Moab Workload Manager to schedule and manage user jobs. Torque is based on OpenPBS, most of the commands for managing your jobs on Blue Waters will be the same as PBS commands. The application launcher (aprun) utility on the Cray system launches applications on compute nodes similar to mpirun on many other systems to launch jobs. Application Level Placement Scheduler (ALPS) take care of job placement and execution of the applications submitted by aprun.

The Blue Waters system also has Cluster Compatibility Mode (CCM), a component of Cray environment to support full Linux compatibility mode. With help of CCM, XE/XK compute node, normally carrying a stripped down operating system, can be turned into a typical node in a standard Linux cluster. This mode is used to run programs on the MapReduce / Hadoop framework.

3.2 Benchmarks Information

In this section we describe the standard and industry benchmarks we have used during our experiments. Some of them are available with the Hadoop distribution while others are developed by the academia or industry.

3.2.1 PI Calculation

PI is a MapReduce program that estimates the value of *PI* using a quasi-Monte Carlo method. This program is available with the Hadoop distribution. *PI* is a purely computational application that employs a Monte Carlo method to estimate the value of *PI*. It is very nearly "embarrassingly parallel": the map tasks are all independent and the single reduce task gathers very little data from the map tasks. There is little network traffic or storage I/O. Detailed information on *PI* program can be found here [22]. BBP is a MapReduce program that uses Bailey-Borwein-Plouffe [39] to compute exact digits of *PI*.

3.2.2 Word Count

Word Count is a MapReduce program that counts the words in the input files. The program counts the occurrences of each word in a large collection of documents. Map emits <word,1> tuples. Reduce adds up the counts for a given word from all map tasks and outputs the final count.

3.2.3 Grep

Grep is a MapReduce program that counts the matches of a regex in the given input. It is helpful in searching a pattern in a file and is a generic search tool used in many data analyses. Each map task outputs lines containing either of the patterns as <regex, 1> tuples. Reduce task adds up the counts and emits <regex, n> tuples. This program is available with the Hadoop distribution.

3.2.4 NNBench

NNBench is a benchmark that stresses the namenode. It is useful for load testing the NameNode hardware and configuration of underlying filesystem. The *NNbench* program is part of Apache Hadoop distribution that can simulate requests for creating, reading, renaming and deleting files on the Hadoop filesystem.

3.2.5 DFSIO

The *DFSIO* program is part of Apache Hadoop distribution to compute the aggregated bandwidh delivered by HDFS. It is a read and write test for the filesystem. The test handles large number of tasks performing read or write operations in parallel. This test run as a MapReduce job, where each map task *i* opens a file to read or write and measures number of bytes transferred and the ececution time for that task. Map tasks followed by a single reduce task for post-processing that aggregates the results from all the map tasks by computing average I/O rate and average throughput for each map task. More information on how to run the benchmark and interpreting the results obtained is explained in [20].

3.2.6 Intel HiBench

Intel's HiBench [4], a Hadoop benchmark suite consisting of both synthetic micro-benchmarks and real world applications such as Sort, WordCount, TeraSort, Bayes, KMeans, NutchIndexing, PageRank, DSFIOE. It can be used as a representative proxy for benchmarking Hadoop applications.

3.2.7 MRBS

MRBS [21] is a comprehensive benchmark suite for evaluating the performance of MapReduce systems. MRBS contains five benchmarks covering several application domains and a wide range of applications that are data-intensive versus compute-intensive or batch applications versus online interactive applications.

4. Experimental Setup

In this section we describe the benchmarking environment along with the configuration setting we used for Hadoop deployment on Blue Waters system and we detail some of the challenges faced in deploying Apache Hadoop software stack on the Blue Waters system.

4.1 Benchmarking Environment

We have used JYC, the Test and Developemnt System (TDS) as well as Blue Waters system for our experiments. Blue Waters has 22640 XE and 4224 XK nodes while JYC consists of 76 XE and 8 XK nodes. The file system is Lustre which is shared across all the nodes.

Resource management and scheduling is handled by Torque. So each job may or may not get different nodes

in the systems and at different network location. We have integrated Yarn with the existing resource management and scheduling software. We use *ccmrun* supported by Cluster Compatibility Mode on Cray systems to properly launch the MapReduce / Hadoop workload on the system.

4.2 MapReduce and Hadoop Settings

We have used an Open Source distribution of Apache Hadoop stack 2.3.0. The node manager resource memory is set to 52 GB which is approximately 80% of memory available on a single compute node. The value for *cpu-cores* is set to 32, virtual core to physical core ratio is set to 2 and virtual memory to physical memory ratio is set to 2. The memory per container is set to 2 GB, therefore we can have 25 number of containers per node. The heap sizes for map task and reduce task are set to 1.6 GB and 3.2 GB respectively. The detailed information on how to set these parameters is available at [27] and we have followed these instructions. We have used same settings on both JYC and Blue Waters systems.

4.2.1 myHadoop

myHadoop [19] is a framework used for configuring Hadoop on traditional HPC resources using the standard job scheduling and resource manager software. User can run Hadoop codes on the HPC resource without having root privileges using myHadoop. It supports a regular non-persistent mode where the local file system on each compute node is used as the data directory for the Hadoop Distributed File System (HDFS), and also a persistent mode where the HDFS can be hosted on a shared file system such as Lustre or GPFS. We have used myHadoop version 2.1.0.

4.2.2 Yarn

MapReduce / Hadoop workloads are excuted on standard Hadoop cluster with the help of resouce management and scheduling entirely handled by Hadoop framework. In contrast, the resouce management and scheduling is always handled by special type of dedicated software or tool Like Torque, Moab or PBS. While, a typical HPC resource has several different users with various types of workloads, Hadoop workload is similar in nature. Each job that runs on HPC system can get different node configurations, can be placed in various toplogy configurations or can get different node types depends upon the type of hardware configuration, available queues and scheduling policies. The changes in the standard Hadoop cluster are very rare in terms of node configuration or node placements in the topology. We have integrated Yarn [37] with the Torque scheduler that is currently available on the system.

4.3 Challenges On Blue Waters

- **Scheduler:**
  User jobs on supercomputing systems are typically managed by a job management system and a resource manager. The Blue Waters system uses TORQUE Resource Manager integrated with the Moab Workload Manager to schedule and manage user jobs. Apache Hadoop stack comes with its own job scheduler, YARN (Yet Another Resource Negotiator). YARN expects to monitor and manage the nodes of a Hadoop cluster. The version of YARN we used in this paper does not integrate with existing workload and resource managers. This is an inherent conflict in how YARN and Blue Waters managers operate. MyHadoop works around this conflict as follows:
  - A regular job is submitted to the existing job scheduler
  - The list of nodes provided by the job scheduler are then used to create a set of configuration files
  - Using these files, a new (temporary) instance of Hadoop cluster is booted up
  - Hadoop jobs are not submitted to this instance of the Hadoop cluster

  In using this technique, other challenges were also encountered. Timeout values for various components had to be tuned to prevent the boot up process from failing. Another major issue was after a Hadoop job completes and tears down the Hadoop cluster, the BW scheduler failed to detect end of the job. This resulted in the BW scheduler waiting for wall clock timeout instead of terminating after job completes. The tear down process was modified so that BW scheduler could detect and release nodes for other jobs.

- **Scaling to larger node counts:**
  The Hadoop boot up script uses secure shell (ssh) to start Hadoop processes on each node. This is done in serial manner. For small node counts, this completed in a reasonable time. As we scaled to larger node counts (50+), the time to configure Hadoop cluster grew at an unacceptable rate. It is observed that he time taken for boot up is doubled when number of nodes are doubled. The time taken for 25, 50 and 100 nodes are shown in table 1.
  Serial tools exist that implement parallel remote shell. For our purposes, we have used *pdsh* [40] mainly because it is already configured to run on BW. Using pdsh instead of ssh, we noticed a significant improvement in boot up time.

Table 1: Timings of boot up using ssh on 25, 50 and 100 Nodes

Noof Nodes	Time
25	179.497
50	355.244
100	700.06

Table 2: Timings of BBP operations on 19 Nodes

Noof Maps	Computing Size	Time
100	0.5 x 10^6	782.53
100	1.0 x 10^7	3224.59
200	1.0 x 10^7	1806.07

5. Results

In this section we illustrate initial results for a spectrum of benchmarks that might give a broad picture of possibility of using HPC resouce such as Blue Waters for large-scale data analytics. We will perform extensive studies of other benchmarks and applications and look into scalabilty of these benchmarks in terms different Cluster Size as well as Data Sizes in detail in the future. We will also evaluate various possible optimization parameters on the system. We will not be running MapReduce / Hadoop workload on the entire system, instead we are planning on using upto 5% of the XE6 nodes which are more than 1000 nodes. If the results are encoraging we will perform scalability studies upto 2000 nodes for the purpose of this paper.

5.1 Performance Evaluation

In this section, we show the results of different standard Hadoop Benchmarks on the Blue Waters system. We also made sure that the experimental setup almost inline with the setup used in the work done at [30] so that we can have fare comparison of the obtained results. We also present the results that illustrate the impact of different data sizes and node sizes and scalability studies with respect to Data size and Cluster size.

5.1.1 PI Calculation

BBP is a map/reduce program that uses Bailey-Borwein-Plouffe to compute exact digits of *PI*. This program is available with the Hadoop distribution. BBP is a purely computational application that employs a Bailey-Borwein-Plouffe method to estimate the value of *PI*. It is very nearly "embarrassingly parallel": the map tasks are all independent and the single reduce task gathers very little data from the map tasks. There is little network traffic or storage I/O. The results of BBP on 5 nodes is shown in figure 2. It shows that the time taken is decreasing when number of map tasks are increased. BBP is computing 0.5 x 10^6 digits. The time taken on our system is 25% less than the time taken in [30] paper. The results obtained for 19 nodes are shown in table 2. It shows that there is 56% improvement in time taken when number of Maps are doubled for the computation of 1.0 x 10^7 digits.

We also have performance numbers on more numbers of nodes that are encoraging. However we have not presented it here as we have not completed all the runs at this time.

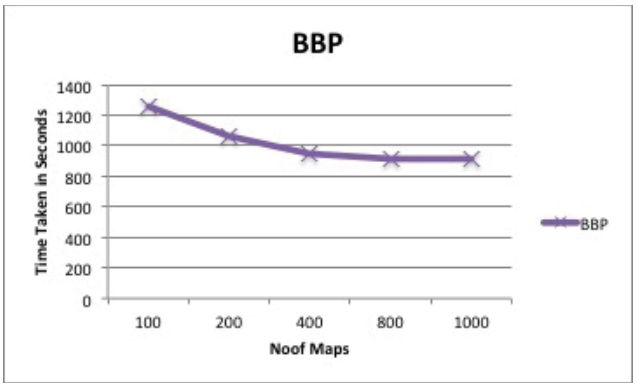


Fig. 2: Performance of BBP on 5 Nodes

5.1.2 Word Count

We have used two datasets for Word Count. One is Wikipedia [29] and other is Freebase [28]. The oroginal size of Wikipedia dataset was small, so we have duppli-cated the dataset few number of times to make the larger dataset of size 105GB. Freebase is an opensource datase released by Google. The size of this dataset is 361 GB. This dataset is a knowledge graph database for structuring human knowledge, which is used to support the collaborative web based data oriented applications. We have used 5 nodes for the performance of Word Count opeartion so that we can compare the results obtained in [30] with our results. The total time taken for wikipedia database is 2719 seconds and 4312 for Freebase database which is little more compare to [30] results. We will investigate further the reason behind this operation.

5.1.3 Grep

We have used Wikipedia [29] and is Freebase [28] datasets for the Text Search operation with datasizes 105 GB and 361 GB respectively. We have used 5 nodes for the performance of Text Search opeartion so that we can compare the results obtained in [30] with our results. The total time taken for Text Search Operation for Wikipedia dataset is 1019 seconds while for Freebase is 2884 seconds. There is more than 50% of reduction in the execution time on the JYC system as compare to results in [30]. We are confident that we will observe similar performance on Blue Waters too. On 20 nodes the time taken is 874 seconds and 300 seconds for Fressbase and Wikipedia dataset respectively. The performance is order of 3.5 magnitute improved with respect to the results obtained on 5 nodes.

5.1.4 TestDFSIO

DFSIO test handles large number of tasks performing read or write operations in parallel. In this test we have used 25, 50, 100, 200, 400 and 500 XE nodes. The total numbers of file written and read were 625, 1250, 2500, 5000, 10000 and

Table 3: Timings of Write and Read operations of DFSIO on 25, 50, 100, 200, 400 and 500 Nodes

Number Of Nodes	Write Oper	Read Oper
25	296.612	334.23
50	372.98	368.72
100	371.3	435.15
200	373.7	487.15
400	374.23	501.1
500	375.92	511.76

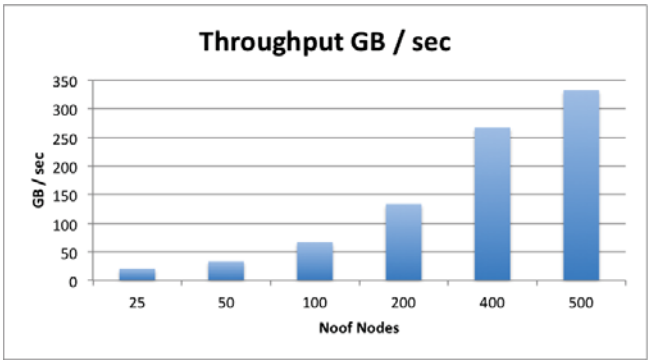


Fig. 3: Performance of DFSIO on on 25, 50, 100, 200, 400 and 500 Nodes

12500 respectively. The time taken for both Write and Read opeartions in mentioned in table 3. For 25 nodes we have observed 21.07 GB/Sec, for 50 nodes 33.51 GB/Sec, for 100 nodes 67.33 GB/Sec, for 200 nodes 133.79 GB/Sec, for 400 nodes 267.21 GB/Sec and for 500 nodes 332.51 GB/Sec. The figure shows that the throughput increases linearly as number of nodes are increased. The average IO rate is 187 MB/Sec for all the five configurations for write operations and varies between 159 MB/Sec to 175 MB/Sec for read opeartions. The throughput obtained is shown in figure 3. The default dfs.blocksize is 128 MB, therefore we have set the Lustre strip size to 128 MB and Lustre stripe count to 160, which is maximum number of OSTs to be used for the *scratch* file system.

6. Related Work

Benchmarking is a de-facto process to measure performance of any given system using a specific operation or set of programs to compare the achieved results with standard measures or other similar systems. Benchmarks are used not only to test but also to measure and to predict the sustained performance of computer system. Benchmarks are also used to reveal their architectural weakness and strong points. Benchmark data can provide valuable insight into the likely behavior of a given system; it may also be used to predict the performance of a new design. Benchmark data on the other hand reflect more specifically how appropriate the given design is for particular set of programs. Benchmarks can be classified according to application classes, such as scientific computing, commercial applications, distributed systems, network services, multimedia applications, and signal processing, etc. It is an important factor for evaluating distributed systems, and extensive work has been conducted in this area. There are various scientific and industry standard performance benchmarking programs available. Some of them are domain specific, some are associated with computer hardware or software systems.

One of the most popular benchmark suite is TPC benchmarks. The Transaction Processing Performance Council (TPC), a non-profit organizatio that defines transaction processing and database benchmarks, and distributes vendor-neutral performance data to the industry. They have several domain specific benchmarks such as TPC-C [5] and TPC-E [6] , an on-line transaction processing benchmark to evaluate online transaction processing (OLTP) performance on various hardware and software configurations, TPC-DS [8] and TPC-H [7], evaluates decision support systems, while TPC-App [9] is an application server and web services benchmark.

These benchmarks are useful in analysing performance of distributed systems, however they are not suitable to evaluate MapReduce framework. The scheduling policies [10], data replication and partitioning policies [11], [12] involve functionalities of microbenchmarks such as grep, word count and sort which are available with standard Hadoop distribution as described in [1].

There are few papers [13] and [14] depicting performance of MapReduce on parallel database systems. In [15], the authors compare MapReduce with parallel database system while in [16] authors study how the job configuration parameters affect the performance of Hadoop. In [17] , authors focus on architectural design issues and possible solutions to improve the overall performance of Hadoop.

In [18], authors discuss about the framework which is strongly based on myHadoop [19] approach to run Hadoop workload on HPC machines and initial results on 33 nodes of Cray XE6 / XK7 system. However none of them have performed detailed studies on the system as massive as Blue Waters, a Cray XE6/XK7 system consisting of more than 22,640 XE6 compute nodes (each containing two AMD Interlagos processors) augmented by more than 4224 XK7 compute nodes (each containing one AMD Interlagos processor and one NVIDIA GK110 "Kepler" accelerator) in a single Gemini interconnection fabric. While the results were obtained in [18] are using in memory for the Cray system, we will be using shared files system, Lustre.

In [23] authors discuss on optimizing nonblocking MPI [26] collective operations to optimize MapReduce and in [24] authors talk about a collective communication library, Harp that can be used to support various applications from HPC to cloud systems. In [25] authors have developed a high performance MapReduce system for the MPI environment that can be used to develop scientific applications in the

molecular dynamics fileld. In this paper, we are not focusing on performance of HPC application that use MPI extensively. However, we will look onto it in the future.

BlueWaters is one of the most powerful supercomputers in the world that provides a HPC platform for scientists and engineers across the country to solve wide range of challenging problems. We want to evaluate how such massive HPC resource can help solving large-scale data analytics, data-mining problems using MapReduce / Hadoop framework.

7. Conclusion and Future Work

There is no official support for Hadoop or related stack on the Blue Waters system. The first and most important challenge in achieving the project goal is to build a working Hadoop stack on the system. We have built a working stack on the Blue Waters system but it is not available to users. The focus of the paper is to obtain benchmarking results and not to provide a stable Hadoop stack on the system. Using the latest version may not be feasible due to the software ecosystem limitations. Therefore, the version works best within the system limitations will be used.

Hadoop works with a share-nothing architecture, where as systems such as BW are share everything designs. Using a shared file system like Lustre may pose challenges. Some workarounds are being investigated but their feasibility on Blue Waters system is unknown at this time. We have integrated YARN with the resource scheduler, MOAB that is available on the system.

Currently, we have only initial results with the existing opensource Apache Hadoop stack [2]. If this does not produce comparable results, we will consider Ohio State University's IB-enabled Hadoop stack [31]. Blue Waters can expose VERBS interfaces over Gemini network using IBoGNI. However, the stability, and compliance of IBoGNI is not well known. If it is stable, this stack will be preferred.

In this paper we have presented initial results of few benchmarks such as PI, Grep, TestDFSIO, NNBench etc. We will be considering few other standard benchmarks such as terasort, contrail bio workload etc. to perform detailed evaluation and analysis of MapReduce framework on Blue Waters. We may consider Intel's HiBench and PUMA benchmark suite and benchmarks if time permits. We will also evaluate MRBS benchmark suite provided the tarball is made available by the developers.

We will perform extensive studies of other benchamrks and applications and look into scalabilty of these benchmarks in terms different Cluster Size as well as Data Sizes in detail in the future. We will also evaluate various possible optimization parameters on the system.

The initial results on the MapReduce / Haddop framework are encoraging and we are confident that the massive traditional High Perfromance Computing resource can be useful in tackling the big-data research challenges and in solving large-scale data analytics, data-mining problems.

8. Acknowledgments

This research is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications. We would also like to thank to the anonymous reviewers and colleagues for their detailed and thoughtful comments and suggestions.

References

[1] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters", in Proceedings of the 6th Conference on Symposium on Opearting Systems Design & Implementation - Volume 6, ser. OSDI 2004. Berkeley, CA, USA

[2] [Online]. Apache Hadoop Available: https://hadoop.apache.org/

[3] [Online]. Blue Waters. Sustained Petascale Computing, NCSA, University of Illinois; 2014 Available: https://bluewaters.ncsa.illinois.edu/

[4] Shengsheng Huang, Jie Huang, Jinquan Dai, Tao Xie, Bo Huang, "The HiBench benchmark suite: Characterization of the MapReduce-based data analysis", ICDEW, 2010, 2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW), 2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW) 2010, pp. 41-51

[5] [Online]. TPC-C. Available: http://www.tpc.org/tpcc/default.asp

[6] [Online]. TPC-E. Available: http://www.tpc.org/tpce/default.asp

[7] [Online]. TPC-H. Available: http://www.tpc.org/tpch/default.asp

[8] [Online]. TPC-DS. Available: http://www.tpc.org/tpcds/default.asp

[9] [Online]. TPC-App. Available: http://www.tpc.org/tpc_app/default.asp

[10] M. Zaharia, A. Konwinski, A. D. Joseph, R. H. Katz, and I. Stoica, "Improving MapReduce Performance in Heterogeneous Environments," in 8th USENIX Symposium on Operating Systems Design and Implementation (OSDI2008), 2008.

[11] G. Ananthanarayanan, S. Agarwal, S. Kandula, A. Greenberg, I. Stoica, D. Harlan, and E. Harris, "Scarlett: Coping with Skewed Content Popularity in MapReduce Clusters," in EuroSys 2011 Conference, Salzburg, Austria, April 2011.

[12] M. Eltabakh, Y. Tian, F. Ozcan, R. Gemulla, A. Krettek, and J. McPherson, "CoHadoop: Flexible Data Placement and Its Exploitation in Hadoop," in 37th International Conference on Very Large Data Bases (VLDB 2011), Seattle, Washington, September 2011

[13] A. Pavlo, E. Paulson, A. Rasin, D. J. Abadi, D. J. DeWitt, S. Madden, and M. Stonebraker. A comparison of approaches to large-scale data analysis. In SIGMOD , pages 165(78. ACM, 2009)

[14] M. Stonebraker, D. Abadi, D. J. DeWitt, S. Madden, E. Paulson, A. Pavlo, and A. Rasin. Mapreduce and parallel dbmss: friends or foes? Communications of the ACM , 53(1):64(71, 2010)

[15] S. Babu. Towards automatic optimization of mapreduce programs. In SoCC , pages 137-142. ACM, 2010

[16] J. Dean and S. Ghemawat. Mapreduce: a flexible data processing tool. Commun. ACM , 53(1):72-77, 2010.

[17] Dawei Jiang, Beng Chin Ooi, Lei Shi, and Sai Wu. 2010. The performance of MapReduce: an in-depth study. Proc. VLDB Endow. 3, 1-2 (September 2010), 472-483.

[18] Scott Michael, Abhinav Thota and Robert Henschel, HPCHadoop: A framework to run Hadoop on Cray X-series supercomputers, *Cray User Group Meeting 2014*

[19] S. Krishnan, M. Tatineni, and C. Baru, "myHadoop - Hadoop-on-Demand on Traditional HPC Resources". Accessed 04-28-2014. http://www.sdsc.edu/ãllans/MyHadoop.pdf

[20] [Online]. Available: http://www.michael-noll.com/blog/2011/04/09/benchmarking-and-stress-testing-an-hadoop-cluster-with-terasort-testdfsio-nnbench-mrbench#testdfsio

[21] Sangroya, Amit and Serrano, DamiÃąn and Bouchenak, Sara, MRBS: A Comprehensive MapReduce Benchmark Suite. Technical Report (RR-LIG-024), LIG Laboratory, University of Grenoble, Feb 2012.

[22] [Online]. Available: http://hadoop.apache.org/docs/current/api/-org/apache/hadoop/examples/pi/package-summary.
[23] Torsten Hoefler, Andrew Lumsdaine, and Jack Dongarra. 2009. Towards Efficient MapReduce Using MPI. In Proceedings of the 16th European PVM/MPI Users' Group Meeting on Recent Advances in Parallel Virtual Machine and Message Passing Interface, Matti Ropo, Jan Westerholm, and Jack Dongarra (Eds.). Springer-Verlag, Berlin, Heidelberg, 240-249.
[24] Zhang, Bingjing, Yang Ruan, and Judy Qiu. Harp: Collective Communication on Hadoop. Technical Report Indiana University, 2014.
[25] Matsuda, Motohiko, Naoya Maruyama, and Shin'ichiro Takizawa. "K MapReduce: A scalable tool for data-processing and search/ensemble applications on large-scale supercomputers." Cluster Computing (CLUSTER), 2013 IEEE International Conference on. IEEE, 2013.
[26] Message Passing Interface Forum: MPI: A Message Passing Interface Standard.
[27] [Online]. Available: http://hortonworks.com/blog/how-to-plan-and-configure-yarn-in-hdp-2-0/
[28] [Online]. Available: https://developers.google.com/freebase/data
[29] [Online]. Available: http://dumps.wikimedia.org/enwiki/latest/
[30] Min Li, Liangzhao Zeng, Shicong Meng, Jian Tan, Li Zhang, Ali R. Butt, and Nicholas Fuller. 2014. MRONLINE: MapReduce online performance tuning. In Proceedings of the 23rd international symposium on High-performance parallel and distributed computing (HPDC 14). ACM, New York, NY, USA, 165-176
[31] [Online]. Available: http://hibd.cse.ohio-state.edu/
[32] Frank Schmuck and Roger Haskin. 2002. GPFS: A Shared-Disk File System for Large Computing Clusters. In Proceedings of the 1st USENIX Conference on File and Storage Technologies (FAST '02). USENIX Association, Berkeley, CA, USA, , Article 19
[33] [Online]. Available: http://en.wikipedia.org/wiki/IBM_General_Parallel_File_System
[34] [Online]. Available: http://lustre.org/
[35] HPC Scheduling and Job Prioritization. [Online]. Available: https://www.adaptivecomputing.com/wp-content/uploads/2014/06/HPC_Scheduling_White.pdf
[36] TORQUE Resource Manager [Online]. Available: http://www.adaptivecomputing.com/products/open-source/torque/
[37] Vinod Kumar Vavilapalli, Arun C. Murthy, Chris Douglas, Sharad Agarwal, Mahadev Konar, Robert Evans, Thomas Graves, Jason Lowe, Hitesh Shah, Siddharth Seth, Bikas Saha, Carlo Curino, Owen O'Malley, Sanjay Radia, Benjamin Reed, and Eric Baldeschwieler. 2013. Apache Hadoop YARN: yet another resource negotiator. In Proceedings of the 4th annual Symposium on Cloud Computing (SOCC '13). ACM, New York, NY, USA, , Article 5 , 16 pages.
[38] C. Clos, "A Study of Non-blocking Switching Networks", Bell Systems Technical Journal, vol. 32, 1953.
[39] David H. Bailey, Peter B. Borwein and Simon Plouffe, "On the Rapid Computation of Various Polylogarithmic Constants", Mathematics of Computation, vol. 66, no. 218 (Apr 1997), pg. 903 - 913.
[40] [Online]. pdsh. Available: http://code.google.com/p/pdsh/
[41] Kramer, William, Michelle Butler, Gregory Bauer, Kalyana Chadalavada, Celso Mendes, Blue Waters Parallel I/O Storage Subsystem, High Performance Parallel I/O, Prabhat and Quincey Koziol editors, CRC Publications, Taylor and Francis Group, Boca Raton FL, 2015, Hardback Print ISBN 13:978-1-4665-8234-7.
[42] Brett Bode, Michelle Butler, Thom Dunning, William Gropp, Torsten Hoe- fler, Wen-mei Hwu, and William Kramer (alphabetical). The Blue Waters Super-System for Super-Science. Contemporary HPC Architectures, Jeffery Vetter editor. Sitka Publications, November 2012. Edited by Jeffrey S . Vetter, Chapman and Hall/CRC 2013, Print ISBN: 978-1-4665-6834-1, eBook ISBN: 978-1-4665-6835-8

Analytic Query Engine

R. Nazarov[1] **and G. Speegle**[1]
[1]Department of Computer Science, Baylor University, Waco, TX. USA

Abstract— *Data analysts are both blessed and cursed by a large number of tools at their disposal. A metatool that allows data analytics above the tool level can provide numerous benefits, such as logical data independence, elimination of duplicate computation and dynamic data processing. Our Analytic Query Engine (AQE) optimizes Analytic Query Language (AQL) a high level language to provide performance with ease of use and stability in a chaotic market. AQL is a non-procedural query language compiled to Big Data programs. The intermediate results of the queries are stored in a cache and reused when similar queries are issued. The program execution can be altered in the face of concurrent executions.*

Keywords: big data, duplicate execution elimination, adaptive query processing

1. Introduction

Big Data is the technological boom for the 2010s. New tools are constantly introduced and old tools are constantly reconfigured. The Apache Hadoop Ecosystem changed the configuration management with the introduction of YARN and the basic computation engine with the introduction of Tez. Meanwhile, competing open source products range from complete ecosystems such as BDAS [1] to specific tools such as GraphLab [2], which recently changed to a commercial product called Dato. Commercial products range from wrappers of open source products to standalone programs to closed source systems. New programming languages are being developed and old ones are being repurposed. Methods for accessing the data range from SQL-like to Java APIs to special purpose languages.

As a result, users today are faced with the challenge of not only selecting the right tool for their big data application, but they must be concerned about the viability of the tool. Investing the resources to develop quality applications for a tool that may not be supported in the future is a signficant risk in today's dynamic big data world. Access to data and analysis without direct concern about the tool mitigates this risk and allows the user to concentrate on the analysis, not the tool.

Furthermore, historically the concept of *logical data independence* [3] has provided a significant advantage for relational databases over competitors such as hierarchical, network and object-oriented models. Logical data independence separates the logical design from the underlying data storage. Efficient execution is achieved through optimization as opposed to relying on developers. This allows quicker query creation and is a key component to the success of the relational model for over 30 years.

We propose a big data metatool that allows users to access data above the tool level. Basic access is achieved via the *Analytic Query Language*, denoted AQL and introduced in [4], which is a non-procedural description of the results. The challenge for AQL is similar to the challenge faced by SQL in the relational database world – computation must be efficient enough so that the gains from the simplicity of the interface are worth the extra computation time per execution.

This paper presents the Analytic Query Engine, called AQE, which provides three benefits not possible from standalone programs. First, AQE supports a non-procedural query language for big data analytics. The syntax is more simple than the programming APIs used by big data tools and yet allows sophisticated programs not easily incorporated in SQL-like languages. Second, AQE queries are independent of the underlying tools. Thus, if a tool is updated or made obsolete, the query issued by the user does not change. Third, AQE provides optimization not possible for standalone programs. In particular, AQE eliminates redundant executions and adapts to concurrent query execution.

2. AQE Design

AQE translates a high-level query language, AQL, into a series of executable statements. Each statement is an invocation of a big data tool over a set of inputs stored on HDFS. The output is stored as an intermediate file so that it can be used later in the query execution. The final result is returned to the user as a handle which can be used to download the data.

A query is submitted to the system as an AQL statement. The parser translates the AQL into an *intermediate query object*, denoted IQO. Using Metadata, an IQO can be instantiated into an executable command. The code generator accepts a set of executable commands as input and outputs a bash script to generate the results. The bash script is executed and a data descriptor is returned to the user for download or further refinement.

The optimization of the query occurs on the IQO. The system is designed to perform three optimizations. The first generates equivalent orderings of the data elements in the statement. The goal of this phase is to detect orderings which can be performed more efficiently. The second phase, called *reuse optimization* takes advantage of the incremental

nature of data analytics. Reuse optimization determines if a portion of a query has been executed and stored in the data repository. If it has, reuse optimization replaces the execution in the IQO with a reference to the stored data. The third phase is an adaptive query processing. If new data becomes available that will allow more efficient reuse, the query is rewritten to use the new data.

2.1 AQL

AQL is described in more detail in [4]. Here we provide a simple overview and note new extensions to the language. AQL consists of two clauses, the **FROM** clause and the **USING** clause. The FROM clause consists of a comma separated list of data sets. A data set may be either

1) A data set previously stored
2) The result from a collection of system defined operations such as *intersection* and *join*
3) A subquery

The USING clause is optional and contains user-defined commands to operate on the data set of interest. The output of an AQL statement is a data set formatted according to the USING clause or a default format according to the data in the FROM clause if the USING clause is empty.

For this paper, we use the following example query taken from the bioinformatics domain and also appearing in [4]. In time-series data, biologists collect genome information from a test subject over a period of time. It is interesting to note the biological components which are preserved across the experiments. The common genome data is calculated by taking the intersection of all samples. The biological elemnets are found by finding the connected components in the intersection. The genome data is represented as a graph. The AQL statement for finding the common elements after four samples is in Listing 1.

Listing 1: Simple AQL for Finding Common Biological Elements in Four Graphs

```
FROM g0 INTR g1 INTR g2 INTR g3
USING connected_components()
```

One of the most powerful aspects of AQL is the use of subqueries in the FROM clause. This allows composition of operations stored as programs in AQE. For example, MapReduce can perform an intersection over an arbitrary set of data sets in one cycle. Thus, it may be more efficient to perform a MapReduce intersection instead of an in-memory binary intersection over a large number of data sets. Listing 2 shows the same query as Listing 1, but uses subqueries.

Listing 2: AQL with a Subquery for Finding Common Biological Elements in Four Data Sets

```
FROM (FROM g0,g1,g2,g3
      USING intersect() )
USING connected_components()
```

2.2 Overview of Query Processing

AQE uses three distinct objects loaded by users. Obviously, the first object is the actual data. The data is stored into a repository (currently HDFS). The second component is the big data tools managed by the system administrators. In order for a tool to be available, it must also be "loaded" into the system. This loading provides the necessary information for AQE to automatically execute the tool. The third component is the programs written in tool specific languages that implement particular algorithms. Programs are "loaded" similar to tools, except programs are be loaded by users. When loading a program, the user must provide the information needed to execute the program, such as the number of parameters required and the tool needed to execute the program. Each object has an entry in the corresponding metadata table (DataTable, ToolTable, ProgramTable) which stores the required information. The metadata tables are discussed in detail in Section 2.3.

When a query is submitted by a user, an *intermediate query object*, denoted IQO, is generated. Initially, an IQO contains an ordered list of operations, but through an instantiation process, an IQO eventually contains a list of the programs to be executed and the operands for each program. The operands include intermediate results which are generated by one program and used as input by another.

The code generation process translates an IQO into an executable shell script which produces the final results. Effectively, the data, program and tool metadata are combined to create a command for each operation. The executable program uses HDFS to communicate data from one command to the next, by storing the intermediate results as files in the HDFS.

Optimization occurs as part of the instantiation process. Currently there are three aspects for optimization. Each aspect is discussed in its own subsection.

1) Generate equivalent IQOs (Section 2.5)
2) Exploit execution reuse (Section 2.4)
3) Adaptive Query Processing (Section 2.7)

2.3 Metadata

There are three primary metadata tables within the AQE: tools; programs; and data. Data is information stored within a repository. It can be structured or unstructured. Currently, the implementation supports graphs and tables as structured types. Tools are big data systems such as MapReduce[5] or GraphLab [2]. Programs are code written for a particular tool for a particular type of data. In order to dynamically use tools to execute programs on data, we use metadata to define the properties of each entity. Table 1 shows the program metadata, as an example.

There exists a one to many relationship between tools and programs. A program is designed to work with one tool, but it is expected for a tool to have many programs. It is even possible for a tool to have many programs with the same

Attribute	Description
Program_ID	primary key
Program_Name	unique name of program
Operation_Type	program functionality
Tool_ID	Tool to execute program
Commutative	is the program associative and commutative with respect to its operands
Priority	selection order
Minimum_Operands	
Input Data Type	
Result Data Type	
Program Invocation	template of the program invocation
Program Statistics	array of values

Table 1: Program Metadata Table. The operation type is the name of the operation implemented by this program, such as intersection or connected components. Commutative is used to identify programs that can be applied to partial results. The priority provides hints for the optimizer with respect to choosing this program over another with the same operation type. The minimum operands field is used in error checking. The program invocation contains placeholders for operands and parameters to be supplied by the query (e.g., k in k-means). Program statistics can be used by the query optimizer for selecting between multiple programs for the same function.

1) g0 INTR g1 $\rightarrow$ r0
2) r0 INTR g2 $\rightarrow$ r1
3) r1 INTR g3 $\rightarrow$ r2
4) connected_components(r2) $\rightarrow$ r3

Fig. 1: An execution generated to implement Listing 1. This execution performs three binary intersection operations and one connected component operation on the results.

operation type. Likewise, different tools may have programs with the same operation type. We envision a typical system will have orders of magnitude more programs than tools, and similarly orders of magnitude more data items than programs.

2.4 Reuse Optimization

In [6], six systems (see Section 4.2.2 for details) are identified with avoidance of redundant processing as a primary objective while 20 others included it as a secondary objective. For AQE, eliminating redundant processing is called *reuse optimization*. The key to reuse optimization is storing partial results from earlier queries and identifying the operations and operands used to create the results. From Listing 1, there are four partial results generated as shown in Figure 1. The value after the $\rightarrow$ represents the id of the stored results.

Each of these results are stored in the repository (in locations which can be referenced via r0 through r3 respectively) and entries are made in the data metadata about the results. Now consider the next time-series query generated. The user would load the next sample, call it g4 and then submit the AQL statement in Listing 3.

1) intersect(r4,g4) $\rightarrow$ r5
2) connected_components(r5)

Fig. 2: An execution to implement Listing 4. It reuses the data generated by Listing 2.

Listing 3: Simple AQL for Finding Common Biological Subgraph Elements in 5 Graphs

```
FROM g0 INTR g1 INTR g2 INTR g3 INTR g4
USING connected_components()
```

Without reuse optimization, five operations would be performed. However, our reuse optimizer detects the repeated operations. The first three steps in the process would be skipped, and only the final intersection and connected components operation would be performed. Each of the first three items in the execution would be replaced by a single existing data set.

Since operations in the FROM clause are system defined, it is straightforward to determine the possibility of reuse. However, for the USING clause, it is more difficult. Consider the same example, only the user submits the AQL in Listing 2. The subquery is executed and stored in the data repository. In general, extra information is required to use the results of an arbitrary function on a subset of elements to generate the results of that function for a set of elements. In AQE, if the function is associative and commutative with respect to its inputs, then it can be applied to any subset of the inputs first and the remaining inputs later. Thus, executing Listing 2 and subsequently Listing 4, the reuse optimizer finds the results of executing Listing 2 (assume it is r4) and generates the code in Figure 2.

Listing 4: AQL with a Subquery for Finding Common Biological Subgraph Elements in Five Data Sets

```
FROM (FROM
      USING intersect(g0,g1,g2,g3,g4) )
USING connected_components()
```

In addition to obvious operations such as intersection and join, more complex user defined functions can be categorized as associative and commutative. For example:

- Incremental log processing, where aggregations are computed over logged information
- Incremental query processing, where queries are executed over streaming or otherwise increasing data
- Incremental graph processing, where new nodes and edges are added to the graph, but none are removed

It is not reasonable for the AQE to determine if a user-defined function is associative and commutative, so whenever a program is loaded into the system, the user must

provide this information. The default setting is false to limit errors in execution.

2.5 Equivalent Executions

A principal mechanism for query optimization in relational databases is re-ordering operations to determine a more efficient execution plan. For example, performing a join on two tables without common attributes requires significantly more computation than two tables with a referential integrity constraint. For the AQE, equivalent queries are generated from two sources. First, the execution order for operations in the FROM clause can be changed for intersection, union and join. Thus, the AQL in Listing 1 is equivalent to 24 similar, but potentially different queries.

The second source for equivalent executions is the Program metadata. Every program with the same operation type potentially generates an equivalent IQO. A program is valid if it supports the number of operands required by the query and the types of the operands. For example, the MapReduce intersection program takes any number of graphs and returns the edges common to all of them. However, the GraphLab graph intersection program is binary. If the query requires performing the operation on three or more graphs, the GraphLab program will not be used.

2.6 Cost Evaluation

After all equivalent executions have been generated and all reuse optimizations have been performed, a set of IQOs must be evaluated in order for the best program to be executed. Typically, this evaluation requires the use of a cost function. Cost functions for relational databases favor particular characteristics such as the smallest intermediate data sets or least data transmitted to another site. For the predefined operations in AQL, similar optimizations can be performed. However, the problem is significantly more difficult for operations in the USING clause. Such operations are arbitrary programs and determining their properties is not always possible.

Thus, the current implementation of the AQE uses a very simplistic cost function: the cost of the query is the number of program executions to be performed. Thus, the cost of the execution in Figure 1 is 5 and the cost of the execution in Figure 2 is 2. This simple cost function takes advantage of reuse optimization when an operation can be eliminated and can be calculated very quickly. Likewise, since AQE uses HDFS as the data communication mechanism, it is reasonable to assume the transition from one program to the next would be slow. Ties are broken by using the priority attribute in the program metadata. If neither IQO dominates the other in priority, the IQO is selected arbitrarily.

Unfortunately, the simple cost function has a strong bias towards programs which can operate on several data sets at once as opposed to binary operations. If the binary programs can operate primarily in memory, it may be faster to execute a series of such programs as opposed to a slower disk based operation. Also, the simple cost function randomly chooses between two IQOs with the same cost. Using the program statistics for a better cost evaluation is current research.

1) intersect g0, g1 → r0
2) intersect r0, g2 → r1
3) connected_components(r1) → r2
4) intersect g0, g1 → r3
5) connected_components(r3) → r4
6) difference r2, r4 → r5

Fig. 3: Execution Plan Without Optimization for Listing 5

2.7 Adaptive Query Processing

The generated execution is very amenable to *Adaptive Query Processing* (AdQP) [7]. In AdQP, the execution of a query can be dynamically modified during execution. Since AdQP can be applied more easily at materialization points and our execution creates a materialization point after every command, then AdQP should be exploited to improve performance.

In particular, reuse optimization should be considered when a partial result needed in a query is generated by another query or earlier in the execution of the query. For example, consider the related query of finding all of the biological components that are lost when gene data g2 is included with g0 and g1. The AQL in Listing 5 produces those results.

Listing 5: The Biological Elements lost when g3 is Included in the Data

```
FROM (FROM g0 INTR g1   INTR g2
        USING connected_components() )
      MINUS
        (FROM g0 INTR g1
         USING connected_components())
```

The execution plan without optimization is in Figure 3.

Obviously, line 1 and line 4 in Figure 3 are redundant computations and one of them should be eliminated. While it is possible to detect this redundancy explicitly within the optimization process, it is more general to use AdQP. The basic template for AdQP [7] consists of four components: 1) monitoring; 2) analysis; 3) planning; and 4) execution.

The monitoring component for AdQP for reuse optimization involves the metadata. Once the execution of a program is complete, a new entry is available in the Data table. The insertion of this data triggers the analysis for all existing execution plans. In this example, the reuse optimizer detects that line 4 now exists as a precomputed result. The planning phase removes line 4 and the reference to its results is replaced with a reference to r0. This optimization is exactly the same as with the current reuse optimization. The simple

1) intersect g0, g1 → r0
2) intersect r0, g2 → r1
3) connected_components(r1) → r2
4) connected_components(r0) → r4
5) difference r2, r4 → r5

Fig. 4: Execution Plan With Adaptive Query Processing for Listing 5

```
hadoop jar /hduser/.../aql.jar -D size="##"
      hdfs:/data/input/ hdfs:/data/results/
mpiexec /hduser/graphlab/conn_component
      --graph=hdfs:/data/results
```

Fig. 5: Example user commands to find the common biological sub-elements in time series data. The ## would be replaced with the number of graphs in the set.

cost function evaluates the new plan as more efficient than the previous plan, and the execution dynamically changes to Figure 4.

AdQP provides better query performance in light of concurrent executions by other queries. For example, if the intersection of g0 and g1 was performed by another query, AdQP would find the updated metadata and modify the execution plan accordingly. Also consider the situation where the amount of available space for cached results is exhausted. It is possible the precomputed data required for an execution is deleted from the repository. In this case, the execution plan would be modified to create the lost data, possibly adding executions to the plan and therefore increasing the cost. If the cost increases, then rejected IQOs should be reevaluated based on the current state of the execution (analysis). If a different IQO is now more efficient, its execution should continue from this point forward (planning).

It should be noted that as of the writing of this paper, the AdQP is not fully implemented in the current AQE.

3. AQE Implementation

The current version of AQE supports loading tools, program and data via a command line interface. Currently, there is no support for distinct users, so all users have all privileges. AQL statements are translated into computations via IQOs with reuse optimization and equivalent executions described in this paper. A debug mode outputs executable scripts as opposed to producing the output.

3.1 Command Parsing

AQE uses Flex [8] to parse AQL into tokens and Bison [8] to generate IQOs. An IQO consists of an ordered list of commands. A command consists of a generic operation (such as "intersection"), a list of operands associated with the command, a program to implement the command, and an identifier for the output. Every command can convert itself into an executable statement using metadata. During query processing, operands are translated from user defined identifiers into HDFS file locations, including operands which have not yet been created. Intermediate HDFS file names include the query identifier to avoid conflicts.

After the IQO is generated, the semantic analyzer detects errors in the AQL statement. For each data item in the statement, the semantic analyzer uses the metadata to determine the existence of the data item and its type. If any data item does not exist, an error is returned. For each operation, the analyzer determines if there exists a program which implements the command on the types of the data items. If no such program exists, an error is returned.

3.2 Execution Examples

In order to show the benefits of reuse optimization, we conducted an experiment on time series data. Given a series of genetic data samples, finding the common biological sub-elements is an interesting question. Given that the user knows the number of samples will grow over time, it is reasonable for the user to select a MapReduce style intersection program that can support any number of data sets. Thus, after performing the experiment and extracting the data, the user would execute something similar to the two commands in Figure 5.

The corresponding AQL statement is similar to Listing 1, but extended to the number of graphs in the experiment. Note that Figure 5 requires the user to be aware of the syntax of all of the tools and the location of the data required for the analysis, while the AQL statement allows the user to only be aware of the user-defined names of the tools and the data.

A series of 9 graphs are used in the experiment. The programs were executed on an Intel i7 3.4 GHz quad core with 32 GB RAM running a 64 bit Ubuntu 12 virtual machine. Table 2 shows the time required for the execution of the commands and the AQL statement as the number of experiments increases. The data used for the experiment is the same as in [4]. Each graph is around 500MB in size. Table 2 discusses the results.

4. Related Work

4.1 Adaptive Query Processing – AdQP

Adaptive Query Processing [7] or AdQP, allows dynamically changing the execution plan for a query during execution. As such, there are four distinct phases: monitoring; analysis; planning and execution. The monitoring phase detects changes to the environment which may require a change in the execution plan. For example, a column may have a data distribution widely different from the distribution used in query optimization or a site in a distributed computation may fail. The analysis phase determines if a monitored change impacts the query execution. A site failure would impact a query requesting data from that site, but not a

Number of Graphs	User Commands	AQL
2	96.7	98.6
3	87.8	64.0
4	103.6	58.4
5	140.0	59.5
6	177.6	75.2
7	188.1	174.5
8	195.4	237.0

Table 2: Comparison between optimized AQL and direct user execution of tools. The total time is in seconds. Under AQL, the decrease in the time required for finding the common sub-elements for three sets compared to two sets is due to the decrease in the size of the input. The intersection of the first two sets is much smaller than the original set. Under AQL, the increase in time after 5 sets is due to the number of orderings considered by the optimizer. Improving this portion of the optimizer is ongoing work.

query that did not access it. The planning phase creates an alternative query execution, such as accessing data from an alternative site. The alternative may require modifications to other parts of the query, including parts that have already been executed. Once all of the adaptions have been planned, they are executed.

4.2 Big Data

The number of big data systems, both commercial and academic, is growing rapidly. For example, as of March 18, 2015, there were 184 papers published since 2014 in the ACM Digital Library with "big data" used as a keyword. We focus here on the results we have found which are the most similar to this work.

4.2.1 Orca

Orca [9] is an SQL query optimizer for big data applications. Orca uses a collection of logically equivalent execution plans which support recursive queries, similar to the IQOs used by AQE. Likewise, Orca considers alternative implementations of operations as well as different execution orders, as AQE does. There are several differences as well. As a complete optimzation product, Orca has a more robust cost function than AQE. Also, Orca supports optimzation over diverse computation architectures (e.g., MPI versus Hadoop) by separating query optimization from the database and abstracting the operating system into their GPOS. Orca does not appear to support adaptive query processing nor user defined analytic operations.

4.2.2 Reducing Replicated Operations

In [10], ideas similar to our reuse optimization are applied to HiveQL, including user defined functions (UDF). Hive computations are translated into MapReduce programs which store intermediate results in the HDFS. This work treats those intermediate results as "opportunistic materialized views" which are used in the construction of the execution plan. A UDF is modeled as a "grey box" with local functions providing semantic information. The local functions are added as annotations to the code. The local functions provide more semantic information than our classification of UDFs as associative and commutative, and therefore provide greater opportunity for optimization. However, our classifications are easier to provide. As such, we provide a "darker grey box" view of UDFs than [10].

Additionally, [6] cites several systems that have a primary goal of avoiding redundant processing. These are identified as MRShare [11], ReStore [12], Sharing Scans [13], Silva et al [14], and Incoop [15]. In MRShare and Sharing Scans, the input operations are shared. Restore, Silva and Incoop store intermediate results, similar to our system, but they do not consider UDF semantics.

4.3 Analytic Languages

We defined an analytic language as a high-level design to process data via parallel computation. Under this definition, Java within Hadoop's MapReduce is an analytical language. In this section, we describe three languages similar to AQL – Pig Latin [16], HiveQL [17] and SCOPE [18].

Scope [18] is used in Microsoft's large scale data analytic engine. It contains both an SQL-like interface and a scripting interface based on C# components. In both cases, Scope is declarative and uses an optimizer to determine the most efficient computation. Scope allows user defined functions, but limits them to extensions of specific C# functions. Similarly, Scope does not use AdQP.

Pig Latin [16] is a data flow language that allows for optimization of operations. It supports data manipulation operations such as filtering and grouping while also providing programming constructs such as for-each loops. Pig Latin is an excellent tool for writing big data analytics, and storing Pig Latin programs within the AQE would allow very powerful operations. However, Pig Latin does not support reuse optimization or AdQP.

HiveQL [17] is an SQL-like language for Hive, including both DDL and DML. It supports limited UDFs defined in Java and custom map-reduce scripts. The compiler for HiveQL is very similar to the compiler for AQL, including optimization over predefined operations such as joins. The execution is a DAG of operations, allowing greater parallelism than AQE's linear execution plan. However, HiveQL does not perform AdQP nor take advantage of reuse optimmization.

5. Conclusion and Future Work

AQE is a proof-of-concept implementation of a high-level metatool for big data analytics. It allows users to analyze data via logical data independent queries in AQL without concern over the tools used for executing the analysis.

Furthermore, AQE provides three optimizations of AQL statements:

1) Reordering operations to find all possible optimizations
2) Reusing previously computed results
3) Adapting queries to incorporate concurrent results (in progress)

The early results of the implementation are encouraging with significant time savings over manual query construction. However, the time spent on query optimization must be improved to provide benefits for complex queries.

Reuse optimization for associative and commutative operations is very powerful. However, there are two possibilities we want to consider. First, we want to determine the kinds of reuse optimization that can be performed on programs that are not associative and commutative. In particular, incremental functions could be exploited. We also want to investigate the logical functions used in [10].

The cached results from previous queries must be managed. Eventually, the space for cached results will be exhausted, and some victim selection must be used to allow new results to be stored. Victim selection is complicated by reuse optimization, as cached results may be used by currently executing programs. It is even possible for a deadlock to arise. Consider the case where two queries, q0 and q1, are using two previously generated results, r0 and r1. Assume r0 and r1 consume all available space. Query q0 must wait for q1 to finish in order to remove r1 from the cached space and generate its output. Query q1 must wait for q0 to finish for the same reason. A simple solution is to abort one of the queries and remove its cached results. AdQP would then rewrite the aborted query.

AdQP provides the opportunity to modify the execution plan based on the environment and the characteristics of the program. For example, if the program execution requires more memory than is currently available, a less memory intensive tool might be a better choice for a later execution. The monitoring would be the time required for the program to execute. The analysis would be the time required (or, as sometimes happens with tools, if an error is generated). The planning would be the selection of a different tool for the next occurrence of that command (or the re-execution in case of an error).

References

[1] M. Franklin, "Making sense of big data with the berkeley data analytics stack," in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, ser. WSDM '15. New York, NY, USA: ACM, 2015, pp. 1–2. [Online]. Available: http://doi.acm.org/10.1145/2684822.2685326

[2] Y. Low, D. Bickson, J. Gonzalez, C. Guestrin, A. Kyrola, and J. M. Hellerstein, "Distributed graphlab: A framework for machine learning and data mining in the cloud," *Proc. VLDB Endow.*, vol. 5, no. 8, pp. 716–727, Apr. 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2212351.2212354

[3] A. Silberschatz, H. Korth, and S. Sudarshan, *Database Systems Concepts*. McGraw-Hill, 2010.

[4] G. Speegle and E. Baker, "Integration of big data components for nosql problems," in *Proceedings of the 2014 International Conference on Advances in Big Data Analytics*, 2014, pp. 128–134.

[5] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008. [Online]. Available: http://doi.acm.org/10.1145/1327452.1327492

[6] C. Doulkeridis and K. NØrvåg, "A survey of large-scale analytical query processing in mapreduce," *The VLDB Journal*, vol. 23, no. 3, pp. 355–380, June 2014. [Online]. Available: http://dx.doi.org/10.1007/s00778-013-0319-9

[7] A. Gounaris, E. Tsamoura, and Y. Manolopoulos, "Adaptive query processing in distributed settings," in *Advanced Query Processing - Volume 1: Issues and Trends*, B. Catania and L. Jain, Eds. Springer, 2013.

[8] J. Levine, *Flex & Bison*. " O'Reilly Media, Inc.", 2009.

[9] M. A. Soliman, L. Antova, V. Raghavan, A. El-Helw, Z. Gu, E. Shen, G. C. Caragea, C. Garcia-Alvarado, F. Rahman, M. Petropoulos, F. Waas, S. Narayanan, K. Krikellas, and R. Baldwin, "Orca: A modular query optimizer architecture for big data," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '14. New York, NY, USA: ACM, 2014, pp. 337–348. [Online]. Available: http://doi.acm.org/10.1145/2588555.2595637

[10] J. LeFevre, J. Sankaranarayanan, H. Hacigumus, J. Tatemura, N. Polyzotis, and M. J. Carey, "Opportunistic physical design for big data analytics," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '14. New York, NY, USA: ACM, 2014, pp. 851–862. [Online]. Available: http://doi.acm.org/10.1145/2588555.2610512

[11] T. Nykiel, M. Potamias, C. Mishra, G. Kollios, and N. Koudas, "Mrshare: Sharing across multiple queries in mapreduce," *Proc. VLDB Endow.*, vol. 3, no. 1-2, pp. 494–505, Sept. 2010. [Online]. Available: http://dx.doi.org/10.14778/1920841.1920906

[12] I. Elghandour and A. Aboulnaga, "Restore: Reusing results of mapreduce jobs," *Proc. VLDB Endow.*, vol. 5, no. 6, pp. 586–597, Feb. 2012. [Online]. Available: http://dx.doi.org/10.14778/2168651.2168659

[13] P. Agrawal, D. Kifer, and C. Olston, "Scheduling shared scans of large data files," *Proc. VLDB Endow.*, vol. 1, no. 1, pp. 958–969, Aug. 2008. [Online]. Available: http://dx.doi.org/10.14778/1453856.1453960

[14] Y. Silva, P.-A. Larson, and J. Zhou, "Exploiting common subexpressions for cloud query processing," in *2012 IEEE 28th International Conference on Data Engineering (ICDE)*. IEEE, 2012, pp. 1337–1348.

[15] P. Bhatotia, A. Wieder, R. Rodrigues, U. A. Acar, and R. Pasquin, "Incoop: Mapreduce for incremental computations," in *Proceedings of the 2Nd ACM Symposium on Cloud Computing*, ser. SOCC '11. New York, NY, USA: ACM, 2011, pp. 7:1–7:14. [Online]. Available: http://doi.acm.org/10.1145/2038916.2038923

[16] C. Olston, B. Reed, U. Srivastava, R. Kumar, and A. Tomkins, "Pig latin: A not-so-foreign language for data processing," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '08. New York, NY, USA: ACM, 2008, pp. 1099–1110. [Online]. Available: http://doi.acm.org/10.1145/1376616.1376726

[17] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff, and R. Murthy, "Hive: A warehousing solution over a map-reduce framework," *Proc. VLDB Endow.*, vol. 2, no. 2, pp. 1626–1629, Aug. 2009. [Online]. Available: http://dx.doi.org/10.14778/1687553.1687609

[18] R. Chaiken, B. Jenkins, P.-A. Larson, B. Ramsey, D. Shakib, S. Weaver, and J. Zhou, "Scope: Easy and efficient parallel processing of massive data sets," *Proc. VLDB Endow.*, vol. 1, no. 2, pp. 1265–1276, Aug. 2008. [Online]. Available: http://dx.doi.org/10.14778/1454159.1454166

Lightning Sparks all around: A comprehensive analysis of popular distributed computing frameworks

B. Meszaros[1], P. Harsh[1], and T. M. Bohnert[1]
[1]InIT, Zurich University of Applied Sciences (ZHAW), Winterthur, Zurich, Switzerland

Abstract— *For performing Big Data processing, special frameworks need to be used. For a company that wants to start processing Big Data, a multitude of these frameworks presents itself, each with its specific advantages and disadvantages. It can become difficult to make the right choice. This paper provides a short overview over some of these frameworks, each for its specific field. After an evaluation that compares certain aspects of these frameworks with each other, the best fitting framework - Apache Spark - is chosen for a series of benchmark tests.*

Keywords: Big Data processing, MapReduce, Hadoop, Spark

1. Introduction

Big Data is becoming an essential part in many technology companies. The increasing generation of data in social media, science research, economics, et cetera leads to a higher demand for the computer resources that can handle it. While in 1999, the overall data creation amounted 1.5 exabytes [1], this amount increased to 1227 exabytes in 2010 and 2817 exabytes in 2012 [2]. Therefore, many companies try to cut a slice off for their benefit. According to Gartner, they "found that 73 percent of respondents [of their survey] have invested or plan to invest in big data in the next 24 months" [3]. On the other hand, they also warn companies from collecting all raw data in a so-called Data Lake without processing it. This kind of data collection can be unfruitful for many customers because ultimately, "the majority of business users lack this level of sophistication or support from operational information governance routines [to extract information from the Data Lake]" [4].

This is the reason why nowadays, big data processing is becoming a more and more important topic in many fields of computer science. Several methods of processing have been developed, each with its up- and downsides [5]. In order to make a system comply with the current demands, it has to be scalable, preferably horizontally so that the costs can be kept low [6]. Horizontal scaling poses its very own challenges to the software developer. This has led to an important pattern called MapReduce which handles some of these challenges in a standardised way [7]. Once developed by Google, it has been published in 2004 and implemented in several OpenSource frameworks amongst which Hadoop is still the most popular [8].

This paper shall evaluate several of these OpenSource frameworks in respect to their performance at handling a compute-bound task creating a keyword index with word frequencies from a huge data file. This includes elaborating the ideal settings of aforementioned framework for the task at hand.

The rest of the paper is organised as follows. Section 2 reviews some of the currently existing frameworks aimed at processing big data including the key factors that need to be taken into account when evaluating switching to either of the presented platforms. Based on this information, a framework is chosen whose performance shall be evaluated. Section 3 shows the approach that was taken for the performance benchmarks conducted with the chosen framework. This knowledge is presented and discussed in section 4. Finally, section 5 concludes the paper and shows possible further research.

2. Related Work

There are several OpenSource frameworks that at least partly employ the MapReduce programming model, others utilise different approaches. The ones referred to in this paper are Hadoop (MapReduce)[1], HaLoop[2], Storm[3], Giraph[4] and Spark[5]. While Hadoop (MapReduce) is the most established framework of those taken into account of this paper, all other frameworks have a right to exist in their own way. An overview over each framework's qualities shall give the reader a preliminary insight.

There are several main attributes that need to be considered when it comes to evaluating the right framework for a particular problem. The following are the most important ones that characterise the main points of each framework: Scalability, Data I/O performance, Fault tolerance, Real-time processing, Supported data size, Iterative task support, Data caching (in memory or on disk), the Environment it is executable in and the Programming languages it does support.

[1]Apache Hadoop, http://hadoop.apache.org [accessed: 2015/04/15]
[2]HaLoop, https://code.google.com/p/haloop [accessed: 2015/04/15]
[3]Apache Storm, http://storm.apache.org [accessed: 2015/04/15]
[4]Apache Giraph, http://giraph.apache.org [accessed: 2015/04/15]
[5]Apache Spark, https://spark.apache.org [accessed: 2015/04/15]

2.1 Frameworks

An introduction to the frameworks will be provided first so they can be referred to in the following part with the properties' description.

2.1.1 Hadoop (MapReduce)

Hadoop MapReduce is the best-known OpenSource implementation of MapReduce. This technology makes heavy use of distributed computing on a cluster of low-end computers which accounts for its horizontal scalability. As a consequence, a distributed filesystem, Hadoop Distributed File System (HDFS)[6], based on Google's distributed file system Google FileSystem (GFS)[7], was developed for a quick access to files across the cluster. Integrated into this filesystem is a replication algorithm which by default commits three copies of its content over the network, one of them off-site, for fault-tolerance.

MapReduce is a batch-processing method which makes it unusable for real-time computing. However, since version 2, Hadoop can cooperate with other processing frameworks that therefore can similarly benefit from HDFS and possibly implement real-time processing capability on their own.

2.1.2 HaLoop

HaLoop was developed on top of Hadoop with the objective of enabling it to handle iterative jobs. While Hadoop's MapReduce cannot deal with loops efficiently, HaLoop's implementation was extended with a Loop Control block. Additionally, a Caching and an Indexing block were introduced. Caching happens on the local disk device which noticeably reduces network traffic. Several stop conditions can be defined for the loops and an intelligent loop controller not only assures a loop execution close to the data – possibly on the same physical machine –, but also the prevention of iterations if the same loop had already been executed for the same input data.

Compared to Hadoop, a speed-up factor of 1.85 has been calculated [9].

2.1.3 Storm

Storm was implemented specifically with stream processing in mind. This was achieved by in-memory computing and a high scalability. Storm can be re-balanced on the go. So-called spouts and bolts are the names of data sources and sinks respectively. Bolts can be connected to several spouts or other bolts. An important feature of Storm is its failure tolerance. It is guaranteed that not a single input will go unprocessed. Generally, these inputs come as streams of small data entities such as integers or byte arrays. One of Storm's qualities is its remarkable speed. Its latency is well below comparable frameworks like Spark [10].

2.1.4 Giraph

Giraph was written as a graph processing framework to be run on Hadoop. It can be used for general processing such as matrix and vector calculations. As opposed to underlying Hadoop, Giraph can run iterative jobs. These iterations are called supersteps in the Bulk Synchronous Parallel (BSP) model [11]. BSP is based on asynchronous parallel processing in independent nodes which have a value each and whose only communication is through value-laden edges.

2.1.5 Spark

Spark was designed as a fast alternative to Hadoop's MapReduce with a general programming model. Such a speed-up could be provided primarily with in-memory computing. In-memory caching is possible in Spark. Stream processing is facilitated with the built-in framework Spark Streaming[8] which uses micro-batching for providing near-real-time results. Engineered by the Algorithms, Machines, and People Lab (AMPlab) at the University of California in Berkeley, it forms a part of the Berkeley Data Analytics Stack (BDAS) together with Mesos[9], Tachyon[10], Spark SQL[11], etc. Spark can be deployed as a stand-alone system, on top of Mesos or as a part of Apache Hadoop. Its main advantages over latter framework are its in-memory processing, caching and not being restricted to a specific programming paradigm (i.e. MapReduce).

Although it has to be remarked that several frameworks can be run on Hadoop as well, with the only difference that they are not bundled with Hadoop from the start.

2.2 Properties

The following paragraphs shall give a basic angle of each of these characteristics including a short analysis of each framework's performance in them. Table 1 summarises these findings.

2.2.1 Scalability

Scalability is one of the major assets of distributed computing in the cloud [12]. While vertical scalability is expensive and feasible just up to a certain extent because of limited technological advances, horizontal scalability provides a virtually inexhaustible resource for a low price [6].

In respect to scalability, all frameworks meet the requirements. While clusters of Hadoop (v1.2.1) can run "on

[6]Hadoop Distributed File System, http://wiki.apache.org/hadoop/HDFS [accessed: 2015/04/15]

[7]The Google File System, https://research.google.com/archive/gfs-sosp2003.pdf [accessed: 2015/04/15]

[8]Spark Streaming, https://spark.apache.org/streaming [accessed: 2015/04/15]

[9]Apache Mesos, http://mesos.apache.org [accessed: 2015/04/15]

[10]Tachyon Project, http://tachyon-project.org [accessed: 2015/04/15]

[11]Spark SQL, https://spark.apache.org/sql [accessed 2015/04/15]

Table 1: Overview over Frameworks

Framework / Property	Hadoop (MapReduce)	HaLoop	Storm	Giraph	Spark
Scalability	high	supposedly high	fairly high (see paragraph "scalability")	supposedly high	high
Data Throughput	high (batch processing)	higher than Hadoop	high (in-memory)	high for few jobs	high (in-memory)
Fault Tolerance	high	officially not known	high; several methods	claimed to be high	high but expensive
Real-time Processing	none	none	yes, explicitly	none	high
Supported Data Size	huge	huge	stream of small messages	huge	depending on filesystem
Iterative Task Support	none	yes	none	yes	yes
Data Caching (In-memory or on-disk)	none by default	on-disk	partly	in-memory	on-disk, in-memory
Environment of Execution	Hadoop	Hadoop	standalone, Hadoop (YARN)	Hadoop	standalone, Hadoop (YARN), Mesos
Supported Programming Languages	any language	Java (maybe more)	any language	Java (maybe more)	Scala, Java, Python

clusters with thousands of nodes"[12], HaLoop and Giraph are supposed to be able to run on the same amount as they are based on Hadoop. Spark has been tested with 8000 nodes[13]. A quality of Storm is its "rebalance" command which can adjust parallelism on the fly. Yet it is claimed to have limitations when clusters of several thousand nodes are used[14].

2.2.2 Data Throughput

Data throughput always depends on the type of data that is to be processed. Batch processing frameworks usually have the highest throughput but a lower responsiveness. This makes Hadoop MapReduce perform well within its limits. On the other hand, HaLoop has a higher throughput than Hadoop because of its utilisation of local caching which reduces network usage [9]. While Storm is mostly optimised for small chunks of stream data, Spark handles large volumes of data with a higher speed than Hadoop [13]. The main advantage of Storm is its ability of stream processing which sets it apart from all the other reviewed tools which only support micro-batching at best in the case of Spark [14]. Giraph's strength is its ability to process BSP with MapReduce. While its data throughput is high for a single job, it drops dramatically when concurrent jobs are processed. This is due to its heavy memory usage [15].

2.2.3 Fault Tolerance

Fault tolerance is one of the aspects that regularly gets neglected when evaluating cloud computing frameworks [16]. Nevertheless, this is an important aspect as completely fail-safe systems are impossible to realise. In this respect, Hadoop MapReduce performs best because of its distributed file system HDFS which replicates data three times by default, one of which is off-site if possible. While other frameworks can take advantage of this file system as well, Hadoop MapReduce does it natively. Also failed nodes will automatically be deactivated by the master.

For HaLoop, fault tolerance values are officially not

[12]HDFS Users Guide, https://hadoop.apache.org/docs/r1.2.1/hdfs_user_guide.html [accessed: 2015/04/15]

[13]Apache Spark FAQ, https://spark.apache.org/faq.html [accessed: 2015/04/15]

[14]Taylor Goetz, The Future of Apache Storm: Secure, Highly-Available, Multi-Tenant Processing in YARN, http://hortonworks.com/blog/the-future-of-apache-storm [accessed: 2015/04/15]

known[15]. Storm handles failures of nodes in a special way. It ensures that every single input value will be processed[16]. Additionally, dead workers will be restarted automatically and a worker on a dead node will be restarted on a different node[17]. Giraph is claimed to be fault-tolerant, especially if its state is saved regularly after the supersteps[18]. Spark has a high fault tolerance but it comes at a high price which lies in the duplication of disk data. Lost Resilient Distributed Datasets (RDDs), Spark's in-memory data entities, will be re-built after being lost. Because of Spark's high usage of in-memory computing, much processing time can be lost if in-memory data has not been saved for a longer time [17].

2.2.4 Real-time Processing

Real-time processing has become an essential property of data processing in many fields of computation nowadays. Especially Social Networking sites and fraud detection systems are depending on a high performance of real-time computing systems.

Hadoop MapReduce and the systems built on top of it (i.e. HaLoop and Giraph) are not apt for real-time processing because of their batch processing architecture. Storm on the other hand was developed for real-time stream processing. Spark was built as optimisation of Hadoop MapReduce, but its Spark Streaming part makes it successful in real-time computing as well. In all evaluations, Hadoop was outperformed by Spark.

2.2.5 Supported Data Size

The system's purpose always determines which data size has to be supported. A batch processing system therefore has a higher necessity for large data while a stream processing engine needs to be able to handle a large amount of small data chunks. This solidifies in the frameworks' properties.

Hadoop MapReduce and the systems built on top of Hadoop, namely HaLoop and Giraph, support very large files which is provided by HDFS, their underlying filesystem. A drawback of large file support is the lower performance with small files resulting from a large default block size of 64MB. Storm is designed for stream processing which makes its best-handled data: a stream of integers, floating point numbers or byte arrays. Spark as a general processing framework supports HDFS as well which gives it the same advantage as Hadoop-based systems. Yet Spark can access files on any local or network filesystem as well.

[15]HaLoop FAQ, https://code.google.com/p/haloop/wiki/FAQ [accessed: 2015/04/09]

[16]About Storm, https://storm.apache.org/about/guarantees-data-processing.html [accessed: 2015/04/15]

[17]Apache Storm website, Fault tolerant, https://storm.apache.org/about/fault-tolerant.html [accessed: 2015/04/09]

[18]Processing large-scale graph data: A guide to current technology, http://www.ibm.com/developerworks/library/os-giraph [accessed: 2015/04/09]

2.2.6 Iterative Task Support

When it comes to iterative task support, most systems tend to have their own approaches. As the MapReduce paradigm does not imply iterations, Hadoop MapReduce cannot realise them in a performant way. HaLoop was built precisely for the purpose to introduce loop awareness into Hadoop MapReduce. This was achieved and optimised by caching and node locality, provided by a loop-aware scheduler, loop-invariant data caching and caching for fix point verification [9]. Storm does not have any special iterative task capabilities as its data flow is one-way from a spout to one or more bolts. Giraph has a high support for iterative tasks - the iterative execution of supersteps - for its ability to process graphs. Because of Spark's versatility, no particular software paradigm is implied for programming Spark jobs.

2.2.7 In-memory/local Data Caching

Caching data is a major performance booster of a software. There are two types of cache to speak of: in-memory caches in the local RAM and local disk caches. While in-memory caches save the software from having to write all results onto a disk, a disk cache leads to more locality of data so that less network transfer of data blocks is necessary.

Hadoop's architecture is not optimised for in-memory caching. This can be achieved with additional frameworks such as MemcacheD[19] which results in a considerable performance boost [18]. HaLoop operates on three local on-disk-caches that can be turned on or off by configuration. Giraph performs its calculations in-memory. Storm just caches the input data so that in case of an error, it can be re-processed. Spark does the processing in-memory; if the memory is exhausted, a local disk cache will automatically be taken into use.

2.2.8 Environment to Execute in

The more environments a framework can be executed in, the more valuable it is for its user as less learning overhead is necessary if a known system is supported.

Hadoop, meaning its filesystem HDFS and its scheduler Yet Another Resource Manager (YARN)[20], is a de facto standard for distributed processing in the cloud which makes it well supported amongst most frameworks. Hadoop MapReduce, HaLoop and Giraph run on Hadoop. Storm can either be run on Hadoop with its scheduler YARN or in standalone mode. Spark supports Hadoop's YARN additionally to the standalone mode and Mesos[21].

[19]Memcached Website, http://memcached.org [accessed: 2015/04/15]

[20]Apache Hadoop NextGen MapReduce (YARN), http://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html [accessed: 2015/04/15]

[21]Cluster Mode Overview, http://spark.apache.org/docs/latest/cluster-overview.html [accessed: 2015/04/15]

2.2.9 Supported Programming Languages

The programming language to write jobs in for the different frameworks might not be a criterion of high importance but it sheds a light on the versatility of the handled frameworks.

In this respect, Hadoop MapReduce[22] and Storm[23] are the most versatile systems. Both support any programming language which can write to and read from stdin and stdout respectively.

2.3 Evaluation

Based on aforementioned research, we evaluated the frameworks against each other. Storm clearly handles stream processing best but is not usable for batch processing requirements. Hadoop has got remarkable qualities in batch processing but its deficiencies in real-time processing and iterative processing make it less versatile. HaLoop eliminates Hadoop's lack of iterative task support but real-time processing is still no option. Giraph's strong point is graph processing but it cannot compete with other players when more general processing paradigms are required. Spark alone has earned a high mark in our analysis for batch processing, real-time processing and iterative task support. Additionally, its caching behaviour is admirable with co-existing in-memory and on-disk caching. Its limitation of three programming languages is in our view not a major drawback as it can be optimised in a target-specific way by its maintainers. For this reason, further evaluation in this paper will solely be based on Spark.

3. Approach

Spark can be configured extensively in respect to memory usage. The underlying Java Virtual Machine (JVM) can also take independent settings. In order to test Spark's performance, we take a subset of these parameters into account which we think have most influence. As benchmark job type, we choose WordCount - counting the number of occurrences of each word in a text file - because it does not pose too high demands on the computing power of the individual nodes. Instead, it strains the data flow within a Spark cluster as the data throughput versus computing demand ratio is considerably high. The dataset which serves us as input is always a Wikipedia data dump.

In order to detect the configuration's main influences on the computation, the following parameters are of importance:

- number of executors in the cluster
- number of cores per executor
- amount of memory for each executor
- amount of memory for the tasks

[22]Hadoop Programming with Arbitrary Languages, https://rcc.fsu.edu/docs/hadoop-programming-arbitrary-languages [accessed: 2015/04/15]

[23]Apache Storm website, https://storm.apache.org/about/multi-language.html [accessed: 2015/04/15]

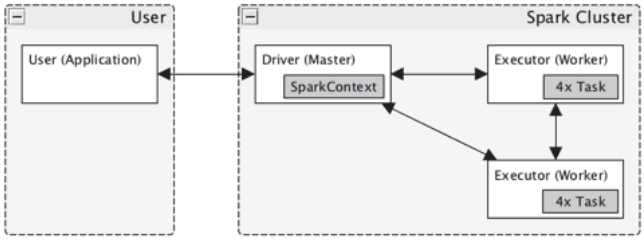


Fig. 1: Spark test environment

- locality of data
- serialisation type of data when caching on disk

In this paper, we are limiting ourselves to varying the first four parameters. With varying the amount of executors as well as the amount of CPU cores per executor, a reasonable benchmark is possible for determining whether (and up to which amount) Spark can achieve a linear performance increase – which would be ideal – when scaled out horizontally. Executing the tasks with different sizes of memory will show which effect the shuffling of data has on the execution speed. We do not take the locality into account because the nodes are all in the same datacenter, i.e. the same physical computer. The data that is used is always stored locally on each node.

Figure 1 gives a schematic overview over the system that is used to perform the jobs. The Spark Cluster consists of a Master Node and two Worker Nodes which both have the same amount of memory, 4GB. Of these 4GB, not more than 3GB can be used by Spark. All nodes are virtualised by the cloud environment OpenStack[24]. The Driver handles the incoming jobs in a SparkContext which partitions them and distributes them among two existing Executors. The Executor is a process running on each Worker Node. It performs the task that it was given by the SparkContext on the Master Node. An Executor task's maximum memory can be configured individually for each job. The workers have got 4 CPU cores which can make them execute up to 4 tasks in parallel. During the Shuffle phase, the Executors communicate with each other for data exchange where (amongst others) serialisation comes into play. The Driver exposes an interface to the user who can transmit new jobs with a script called `spark-submit`.

Every processing job was performed three times in a row and the average was taken so that the result does not rely on one single measurement.

4. Results

The test results that we conducted are summarised in six graphs in figure 2.

In the first part of the tests, we varied the amount of tasks that were performed in parallel by the worker nodes. The

[24]OpenStack website, https://www.openstack.org [accessed: 2015/04/15]

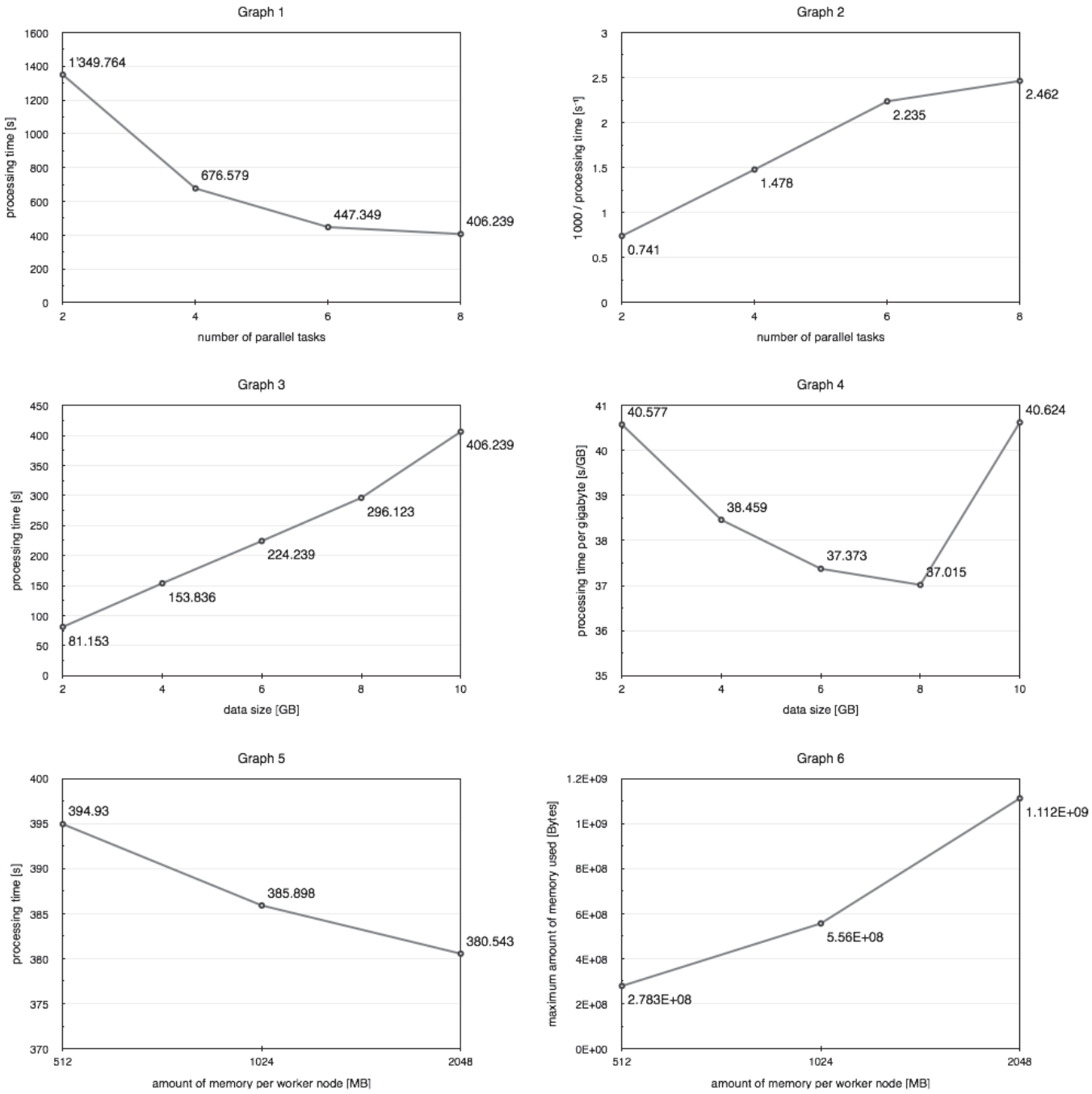


Fig. 2: Testing results

data size was 10GB, there were 2 worker nodes present and the amount of memory per worker node was 2048M. Graph 1 in figure 2 shows the decrease of processing time while the amount of parallel tasks was increased. Graph 2 in figure 2 shows the inverse proportionality of the processing time in respect to the parallel tasks. The performance increase is linear as expected up to 6 parallel tasks. For 8 parallel tasks, the increase is not as steep anymore as expected. We explained this behaviour with the increased disk access which was the bottleneck. This resulted from all virtual machines being executed on the same physical machine accessing the same disk. For this reason, in the regression, we only took the first three values into account. We did a linear regression which resulted in the following values for the equation $y = f(x) = a * x + b$:

$$a = 0.37363$$
$$b = -0.00976$$
$$r = 0.99997$$
$$r^2 = 0.99994$$

The negative value of b can be explained with the statistical dispersion which was always around 6‰. This dispersion probably resulted from other virtual machines engaging the physical computer.

The incline of the line, a, is specific to the individual setup of the executing computer. It is determined by the hardware, the virtualisation software and Spark's configuration.

The correlation coefficient r is very close to 1 which indicates a high linear scalability. For this, the coefficient of determination r^2 remains at a remarkably high level as well. Latter two values affirm that Spark is exceptionally well linearly scalable.

The second test series examined the increase of processing time in respect to an increase of processed data size. The tests were performed with 8 parallel tasks on 2 worker nodes, each with 2048M memory. With this method, we wanted to determine the heterogeneous part in the equation that resulted from a scheduling overhead in Spark. Graph 3 in figure 2 depicts this test. Again, the switch from 8GB of data to 10GB of data lead to an irregular increase of processing time which we already explained in the first test series. According to this graph, there is an overhead of 10s that Spark has to use for scheduling/shuffling activities. Graph 4 in figure 2 shows the inverse graph. It becomes clear that Spark is performing constantly better with the increase in data size up to 8GB, above which the memory write effect becomes visible again. This effect probably could be reduced with the use of SSD disks.

Again, we calculated the hypothetical curve by linear regression up to the value that was not influenced by the disk access latency. The resulting parameters for the aforementioned equation are as follows:

$$a = 35.765598$$
$$b = 10.009988$$
$$r = 0.999983$$
$$r^2 = 0.999966$$

In this measurement, again, the statistical dispersion was linear to the measured values, however, with an average amount of 1.6%, significantly larger. We came to the conclusion that this resulted from the lower runtimes of these calculations which leaves more space for statistical error considering the setup time of the environment.

10 seconds of setup time are obviously too long for real-time processing, but for larger input data, this time does not increase, provided that there is no other bottleneck in the cluster. Also, if real-time stream processing is needed, the framework of choice is Spark Streaming which itself is part of Spark. In this case, the object `StreamingContext` (as an alternative to `SparkContext`) handles the necessary functionality.

The third test was the variation of worker memory which is depicted in graph 5 and graph 6, both in figure 2. As is visible, the processing time did not significantly depend on the amount of memory that each worker node had access to. Yet graph 6 clearly shows that all the memory that was provided had been used by the workers at some point. The low amount of used memory compared to the total memory at disposal, i.e. the scaling by the factor of 0.6, was due to the default fraction of the framework's $spark.storage.memoryFraction$ setting which determines the amount of the provided memory that is to be used for the Java heap. In our view, the Garbage Collector in Java had to be executed less frequently the more memory there was at Java's disposal which accounted for the saved seconds during the benchmark.

5. Conclusion

In the course of this paper, we examined several software frameworks for distributed computing in the cloud. Everyone of them has its own specific field of usage. According to our research, the framework Spark has got the best general processing characteristics because of its versatile structure. For this reason, we decided to perform some benchmark testing with Spark which resulted in two insights into Spark's behaviour: first, the framework's bottleneck on our system was probably the disk whose throughput was too low for Spark's speed. Second, we calculated the scheduling overhead of Spark in a specific situation.

Other insights into Spark were that its setup time for a new task was too long to use it for real-time processing. However, this setup time stayed the same no matter how much data was processed. This makes Spark especially useful for large file processing. On the other hand, Spark has its own built-in quasi-real-time processing engine, Spark Streaming over the object `StreamingContext`, which can handle real-time stream processing.

In our further research, we will perform former tests with SSD disks in order to eliminate the mentioned bottleneck. Additionally, we will make different locality tests, evaluating Spark's behaviour in case that the data is distributed among the network or that the Spark cluster is not fully local. Another interesting topic would be to evaluate Spark's real-time stream processing performance.

6. Acknowledgement

This work was undertaken under the Information Communication Technologies, EU FP7 T-NOVA project, which

is partially funded by the European Commission under the grant 619520.

References

[1] P. Lyman, and H. R. Varian. (2000). How Much Information? 2000. [Online]. Available: http://www2.sims.berkeley.edu/research/projects/how-much-info/

[2] J. Gantz, and D. Reinsel. (2012). The digital universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East. [Online]. Available: http://www.emc.com/collateral/analyst-reports/idc-the-digital-universe-in-2020.pdf

[3] J. Rivera, and R. v. d. Meulen. (2014). Gartner Survey Reveals That 73 Percent of Organizations Have Invested or Plan to Invest in Big Data in the Next Two Years. [Online]. Available: http://www.gartner.com/newsroom/id/2848718

[4] J. Rivera, and R. v. d. Meulen. (2014). Gartner Says Beware of the Data Lake Fallacy. [Online]. Available: http://www.gartner.com/newsroom/id/2809117

[5] C. Ji, Y. Li, W. Qiu, U. Awada, and K. Li. (2012). *Big Data Processing in Cloud Computing Environments*. [Online]. Avaulable: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6428800

[6] E. Singh, and C. K. Reddy. (2014). A survey on platforms for big data analytics. [Online]. Available: http://www.journalofbigdata.com/content/2/1/8

[7] J. Dean, and S. Ghemawat. (2004). MapReduce: Simplified Data Processing on Large Clusters. [Online]. Available: http://research.google.com/archive/mapreduce.html

[8] J. Dittrich, and J. A. Quiané-Ruiz. (2012). Efficient Big Data Processing in Hadoop MapReduce. [Online]. Available: http://dl.acm.org/citation.cfm?id=2367562

[9] Y. Bu, B. Howe, M. Balazinska, and M. D. Ernst. 2010). HaLoop: efficient iterative data processing on large clusters. [Online]. Available: http://dl.acm.org/citation.cfm?id=1920881

[10] R. Lu, G. Wu, B. Xie, and J. Hu. (2014). Stream Bench: Towards Benchmarking Modern Distributed Stream Computing Frameworks. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7027482

[11] T. Kajdanowicz, W. Indyk, P. Kazienko, and J. Kukul. (2012). Comparison of the Efficiency of MapReduce and Bulk Synchronous Parallel Approaches to Large Network Processing. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6406444

[12] Z. Mahmood. (2011). Cloud Computing: Characteristics and Deployment Approaches. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6036733

[13] X. Lin, P. Wang, and B. Wu. (2013). Log analysis in cloud computing environment with Hadoop and Spark. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6823956

[14] M. H. Iqbal, and T. R. Soomro. (2015). Big Data Analysis: Apache Storm Perspective. [Online]. Available: http://www.ijcttjournal.org/Volume19/number-1/IJCTT-V19P103.pdf

[15] J. Xue, Z. Yang, Z. Qu, S. Hou, and Y. Dai. (2014). Seraph: an Efficient, Low-cost System for Concurrent Graph Processing. [Online]. Available: http://dl.acm.org/citation.cfm?id=2600222

[16] M. Schwarzkopf, D. G. Murray, and S. Hand. (2012). The seven deadly sins of cloud computing research. [Online]. Available: http://dl.acm.org/citation.cfm?id=2342764

[17] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica. (2012). Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing. [Online]. Available: http://dl.acm.org/citation.cfm?id=2228301

[18] N. S. Islam, X. Lu, M. Wasi-ur-Rahman, R. Rajachandrasekar, and D. K. Panda. (2014). In-Memory I/O and Replication for HDFS with Memcached: Early Experiences. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7004235

Adaptive Fusion SQL Engine on Hadoop Framework

Shu-Ming Chang[1], Chia-Hung Lu[1], Jiazheng Zhou[1], Wenguang Chen[2], Ching-Hsien Hsu[3], Hung-Chang Hsiao[4], and Yeh-Ching Chung[1]

[1]Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan, R.O.C.
[2]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[3]Department of Computer Science and Information Engineering, Chung Hua University, Taiwan
[4]Department of Computer Science and Information Engineering, National Cheng Kung University, Taiwan

{shuming, lu1227, jzhou}@sslab.cs.nthu.edu.tw,
cwg@tsinghua.edu.cn, chh@chu.edu.tw, hchsiao@csie.ncku.edu.tw, ychung@cs.nthu.edu.tw

Abstract - *As big data becomes popular, data warehouse needs the ability to process massive data fast. Hive is a data warehouse built on Hadoop. It provides SQL-like query language called HiveQL. Although there is a fault tolerance mechanism in Hive, its response time is long. Impala is another SQL engine on Hadoop that is compatible with Hive. Impala's response time is short, but the data processed by Impala is constrained by memory size. Furthermore, it does not provide fault tolerance mechanism. We focus on designing a fusion SQL engine system that combines Hive and Impala. It provides a uniform interface and a fault tolerance mechanism. After the system parses the query from users, it leverages the preprocessed statistics to estimate the memory consumption, and chooses the SQL engine (Hive or Impala) to execute the query automatically. It makes Hive and Impala become complement of each other.*

Keywords: Data Warehouse; SQL Engine on Hadoop; Hive; Impala, Fusion SQL Engine

1 Introduction

Data warehouse is a database designed for reporting, data analysis and decision-making. It is usually updated in batch and it contains large amount of historical data. Apache Hadoop [1] is a platform for the distributed processing of large data sets across a cluster of commodity computers. The data warehouse for big data processing is suitable to be built on Hadoop.

Apache Hive [18] is the first data warehouse software built on Hadoop. It supports SQL-like query language called HiveQL. The HiveQL query will be compiled to several MapReduce jobs to execute. Hive relies on MapReduce, it provides fault tolerance mechanism, but the job will be run for a long time with high latency. Cloudera Impala [2] is a SQL engine on Hadoop and has its own massively parallel processing (MPP) engine. There is no disk writing in Impala except insertion operations; it processes data in memory. The characteristic makes it much faster than Hive. However, Impala does not spill to disk if intermediate results exceed the memory reserved by Impala on a node. The query would fail when any node involved processing the query is out of memory. The problem commonly occurs when joining two large tables.

In our work, we attempt to integrate Impala and Hive into a fusion SQL engine system and accomplish the followings:

- Offer a uniform query interface for Impala and Hive.
- Propose a method to estimate Impala memory usage.
- Automatic failover.
- Better response time.

The rest of the paper is organized as follows. Related work will be described in Section 2. Section 3 introduces the whole system overview and architecture. The estimation method of the query size is presented in Section 4. In Section 5, we show how to estimate Impala memory consumption. Experimental results are shown in Section 6. Section 7 are the conclusions and future work.

2 Related Work

Apache Hadoop [1] is an open-source implementation of Google File System [9] and MapReduce [7]. The Hadoop Distributed File System (HDFS) [17] provides high throughput access to data and is appropriate for large data sets. Apache Hive [18] is the first approach to data warehousing on Hadoop framework. Hive stores data on HDFS and it supports queries in SQL-like query language called HiveQL.

Shark [19] runs on Apache Spark [20] instead of MapReduce, which is a data-parallel execution engine that is fast and fault-tolerant. Cloudera Impala [2] inspired by Dremel [12] has its own massively parallel processing

(MPP) SQL query engine that runs natively in Apache Hadoop.

Ganapathi, et al. [8] propose a prediction framework that guides system design and deployment decisions such as scale, scheduling, and capacity. Gruenheid, et al. [10] use column statistics to improve the performance of HiveQL queries execution in Hive. Shi, et al. [16] propose HEDC++, a histogram estimator for data in the cloud. Okcan and Riedewald [13] focus on processing theta-joins using MapReduce. Hive [4], Shark [11] and Impala are also developing cost based optimizer to enhance the query performance.

There is less work on hybrid SQL engine architecture on Hadoop framework. Similar approach for improving data warehouse on Hadoop is HadoopDB [5]. It combines parallel database and MapReduce framework, and queries multiple single-node databases by using MapReduce as communication layer. To the best of our knowledge, we are the first system to provide fusion SQL engine system on Hadoop framework.

3 Fusion SQL Engine System

The fusion SQL engine system is designed for leveraging two different SQL engines (Hive and Impala) on Hadoop platform to make the average query response time shorter.

3.1 Overview

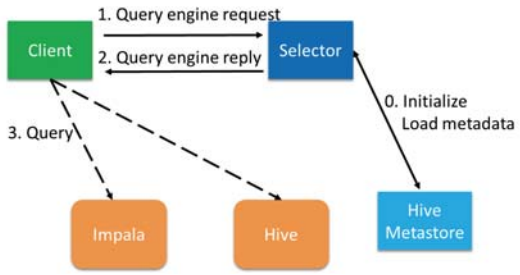


Figure 1: The Workflow of Fusion SQL Engine

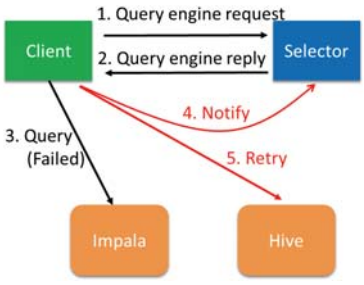


Figure 2: Failover Mechanism

Our system consists of two different SQL engines: Impala and Hive. We will get better performance when submit query to Impala, but query may fail if there is no sufficient memory. Hive can guarantee the execution will always complete but the performance is not as good as Impala. The fusion SQL engine system helps users to automatically determine which SQL engine to use when query arrives.

As shown in Figure 1 and Figure 2, when the system starts, Selector loads the metadata and statistics from Hive Metastore. Selector will use the information later for SQL engine selection decision. When Client gets a query request, Client first asks the Selector which SQL engine should be chosen. The Selector estimates the memory usage of this query and replies to Client. Moreover, our system also supports failover mechanism. As shown in Figure 2, if Impala is first selected as SQL engine, once the query fails, Client will help to perform failover mechanism. It will notify the Selector there is an estimation error and send the query to Hive to get the query result. The Selector logs every failed query in Metastore to prevent from making the wrong estimation again.

3.2 System Architecture

The software stack and architecture are shown in Figure 3. We will introduce them as follows.

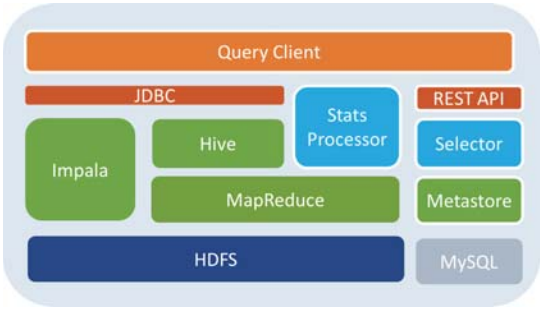


Figure 3: Fusion SQL Engine System Architecture

3.2.1 Client

The Client provides a uniform interface upon different SQL engines. It makes users use different SQL engines to execute query with the same data storage stored in HDFS.

3.2.2 Selector

Selector server loads the metadata and statistics from Hive Metastore when the system starts. Selector will use the information later for SQL engine selection decision. The information is stored in memory for fast access.

Selector estimates memory usage of a query according the execution plan from Impala. Therefore, Selector gets Impala's execution plan in text format via JDBC, and parses it to compute the estimated memory consumption. We will explain the estimation method in Section 4.4.

There is a failover mechanism in Selector. Once the wrong estimation causes "memory limit exceeded" error, Client will notify the Selector to log this query. If Selector gets the same query next time, it will choose Hive directly

instead of Impala to avoid "memory limit exceeded" error.

3.2.3 SQL Engine

There are two SQL engines in the proposed system: Impala and Hive. Impala is faster since it puts data in memory, but it lacks of fault tolerance mechanism. Compared with Impala, Hive runs slower but is reliable. Our proposed system helps users to identify the characteristics of a query by estimating memory consumption.

3.2.4 Metastore

Metastore is one of the components in Hive and used for storing metadata. It uses relational database management system to store information. The first SQL-on-Hadoop is Hive. Most of following SQL engines on Hadoop are compatible with Hive. Metadata contain table schema, partition information, table statistics and column statistics.

3.2.5 Stats Processor

Stats Processor builds statistics data that will be used for Selector. For processing the massive dataset, it is written in MapReduce to speed up the performance. This analysis step does not need to be executed every time after the system starts. Only for the first time users import the data into the data warehouse, the system executes the analysis program to collect some statistics. These statistics data can help to improve the accuracy of estimation.

4 Estimation of the Query Size

In this system, we calculate the approximate memory consumption of a query relying on estimation of query size.

4.1 Preliminary

4.1.1 Terms

- *Predicate*: Each WHERE or JOIN clause is called a predicate.
- *Cardinality*: The number of rows that is expected to return by an operation.
- *Selectivity:* A value between 0 and 1 that indicates the proportion of rows retrieved by one or multiple predicates.

When we estimate the cardinality of this SELECT operation, we need to calculate the selectivity of each predicate denoted as *SEL*(*P*) where *P* is a predicate. We have equation (1) to calculate number of rows.

$$\#Row = Cardinality * Selectivity \quad (1)$$

The selectivity of a predicate affects the result row size. Two methods of calculating selectivity will be described in Section 4.2 and Section 4.3.

4.2 Number of Distinct Values (NDV)

Impala uses the number of distinct values (NDV) to estimate the selectivity of a predicate in current version. The method is proposed by literature [15] and used in System R [6]. It is based on linear and uniform distribution assumption that assumes the number of occurrences of any value in a domain of an attribute is the same. The following are the selectivity formulas based on above assumption for two predicates $p = x$ and $p < x$, p and x denote an attribute and a constant of an attribute respectively.

We have equation (2) and equation (3) to calculate the selectivity.

$$SEL(p = x) = \frac{1}{NDV} \quad (2)$$

$$SEL(p < x) = \frac{x - minval}{maxval - minval} \quad (3)$$

NDV is the number of distinct values of attribute *p*, and *maxval* and *minval* are the maximum and minimum values of attribute *p*.

In many cases, the distribution is neither uniform nor linear. The above formulas would be inaccurate. Thus, for numeric data types, we will use histogram to improve the estimation. For non-numeric data type like string, we will use above NDV method.

4.3 Histogram

A histogram can describe the distribution of attribute values. We can build a single-dimensional histogram to record the value distribution for a single column.

4.3.1 Dimension

Single-dimensional histogram considers the frequency distribution of an individual attribute, and it is based on the attribute independence assumption that there is no correlation among the attributes. Multi-dimensional histogram considers the joint frequency distribution of several attributes. Although multi-dimensional histogram can handle complex join predicate more accurately, the overhead of building it is exponential to the number of columns. In most commercial databases, the single-dimensional histograms are often used.

4.3.2 Equi-width Histogram and Equi-height Histogram

For a traditional equi-width histogram, the internal length of each bucket is the same but the total frequency of each bucket is different. In contrast, for the equi-height histogram, the length of each bucket may be different but the total frequency of each bucket is the same. When the data are skewed, the selectivity estimation from equi-width histogram may be far from real selectivity.

As shown in Figure 4 and Figure 5, we show an example for the equi-width and equi-height histogram, respectively. In this example, most of the customers are 23 years old. If we use equi-width histogram to estimate

SEL(*age* < 22), the error rate of estimation would be larger than that of using equi-height histogram.

The equi-height histogram is proposed by literature [14]. We leverage Piatetsky-Shapiro and Connell's proposed method to build equi-height histograms for every numeric column type and use it to estimate the query result size.

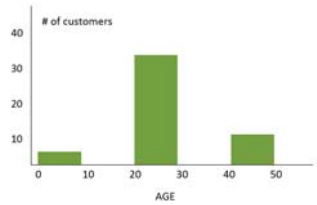


Figure 4: Equi-width Histogram

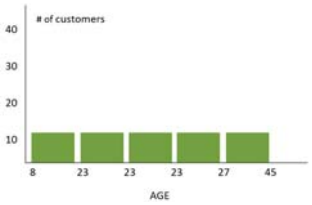


Figure 5: Equi-height Histogram

4.4 Building Equi-height Histogram Using MapReduce

For processing massive data, we build equi-height histograms by using MapReduce. Stats Processor loads the metadata from Metastore and starts up a MapReduce job for each numeric column.

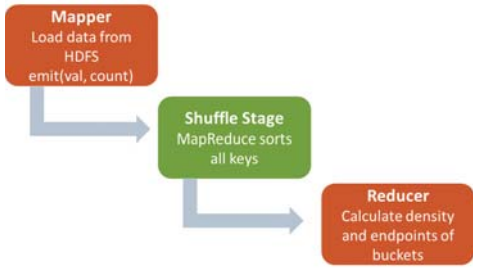


Figure 6: Building Histogram using MapReduce

As shown in Figure 6, the mapper in MapReduce program loads tuples from HDFS and extracts the values from columns. We use the attribute value as output key of mapper to leverage MapReduce shuffle mechanism. It guarantees that the reducer can retrieve keys in order. Therefore, in the reducer, we calculate attribute density value and endpoint of each bucket. Finally, the output of reducer will be stored into Metastore. The endpoints of the histogram in Figure 5 are 8, 23, 23, 23, 27 and 45. The statistics stored in Metastore would help estimate selectivity.

5 Memory Consumption Estimation

5.1 Impala Query Execution

To estimate memory usage of Impala, we need to know how Impala executes our query. In this section, we describe the architecture of Impala and the memory consumption estimation from Impala query execution plan.

5.1.1 The Architecture of Impala

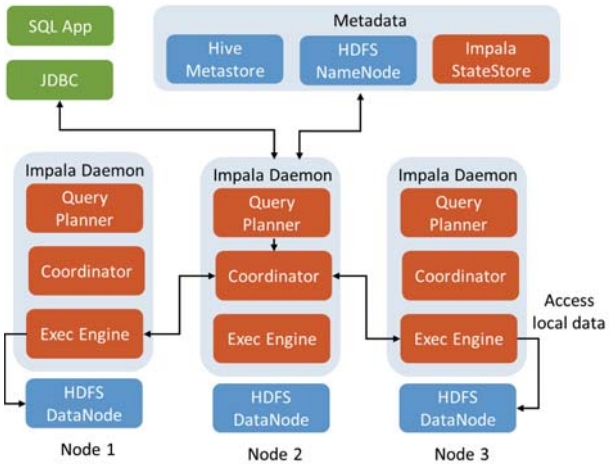


Figure 7: The Architecture of Impala

Figure 7 shows overview of Impala architecture. Impala does not rely on MapReduce to achieve short response time requirement. It directly accesses data on HDFS with a specialized distributed query engine. There are two components in Impala: one is StateStore in the cluster, and the other is Daemon on each HDFS DataNode. StateStore takes responsibility for monitoring the health of each Daemon in the cluster. A SQL application connects to one of Daemons. This Daemon will be the coordinator to generate the query execution plan and coordinate with other Daemons. The exec engines in other Daemons stream the partial results back to the coordinator and the coordinator will response to the user application.

5.1.2 Operations

Select

A select query extracts and filters some data from a table. Impala Daemon scans blocks in HDFS locally and then streams to another node for following operations. Each scan node holds some memory as a buffer. The size of buffer is related to number of blocks in HDFS.

Aggregate Function

The aggregate function like SUM or MIN is performed on separated nodes and then merges the result on an aggregate node.

Join

In general, join operation consumes most of the memory. We focus on the join operation to determine which query can be executed by Impala. Impala uses hash join algorithm to implement the join operation.

As shown in Figure 8, when joining two sets R and S, there are two phases to perform the operation. First, query planner chooses a smaller set R as the input for build phase.

It scans set R and builds the hash table of join attribute. Second, after the hash table is constructed, it scans the larger set S and matches set R by looking up the hash table. The set R must fit into the memory.

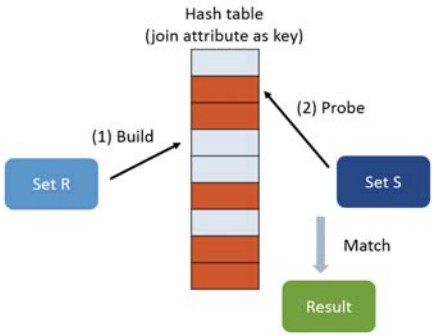


Figure 8: Hash Join

5.2 Estimation from Execution Plan

To estimate the memory consumption for a query, we parse the query execution plan from Impala. Because join operation consumes most of the memory, we focus on it when estimating memory consumption. The hash table is a vector that contains pointers to the table entry data. When the vector is full, it will increase the size automatically. We take the hash table size into account. We set LOAD_FACTOR to 0.75; it means 75% buckets in the hash table contain an entry at least. Each bucket in the vector is a pointer which is 8 bytes in the x64 OS. The estimated memory usage of hash table is calculated in equation (4) and equation (5).

$$\text{mem}_{\text{hashTable}} = 8 * \text{cardinality} * (1/\text{LOAD\_FACTOR}) \quad (4)$$

$$\text{mem}_{\text{join}} = \text{cardinality} * \text{tuple}_{\text{size}} + \text{mem}_{\text{hashTable}} \quad (5)$$

We assume all entries would be distributed to all nodes uniformly, so mem_{join} will be equally divided by the number of nodes that involved in processing the join.

As we mention above, we get the buffer size consumed by HDFS scan node from the query execution plan. There are several plan fragments for a query execution plan. Some of them can be executed in parallel, so we add the estimated memory of those plan fragments to calculate the total estimated memory of a node. The value is used to compare with the memory limit of Impala to determine whether the query can be executed by Impala or not.

6 Experiments

In this section, we evaluate the fusion SQL engine system and compare it with Impala and Hive.

6.1 Experiments Settings

The dataset for our experiments is from AWS [3]. The dataset is a database for the bookstore, which consists of three tables: books, customers and transactions. We generate the dataset as shown in TABLE 1. We build equi-height histograms for the dataset.

We run our experiments on a cluster of 5 nodes. One of these nodes functions as the master node, while others nodes are set up as slaves. Because Impala is only supported under the CDH released from Cloudera, we use CDH 5.0.3 as Hadoop environment. The hardware and software configurations are described in TABLE 2 and TABLE 3.

TABLE 1: Dataset Information

Table	Size	#Records
books	8GB	126,227,290
customers	8GB	105,800,433
transactions	16GB	339,799,026

TABLE 2: Hardware Configuration

Device	Description
CPU	Intel Xeon E5520 @ 2.27GHz x 2 with HT
RAM	24GB
Disk	246GB
Ethernet	1Gbps

TABLE 3: Software Configuration

Name	Version
OS	Ubuntu 12.04.4 LTS x64
Apache Hadoop	2.3.0
Apache Hive	0.12.0
Cloudera Impala	1.3.1
MySQL	5.5.37

6.2 Selectivity Estimation

There are 3 queries in TABLE 4. Q1 and Q2 are scan queries with single selection predicate. Q3 is a join query with a selection predicate. We adjust the constant X to show the variation in estimations.

TABLE 4: Queries for Selectivity Estimation

#	Query
Q1	SELECT count(*) FROM books WHERE price < X
Q2	SELECT count(*) FROM transactions WHERE quantity >= X
Q3	SELECT count(*) FROM transactions JOIN books ON (transactions.book_id = books.id AND books.price < X)

From the results shown in Figure 9, Figure 10, and Figure 11, we can find that the estimation values are close to real case values. We also calculate the mean absolute percentage errors (MAPE) in TABLE 5. All MAPE values are under 1.6%.

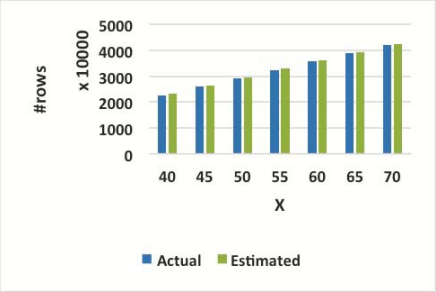


Figure 9: Selectivity Estimation for Q1

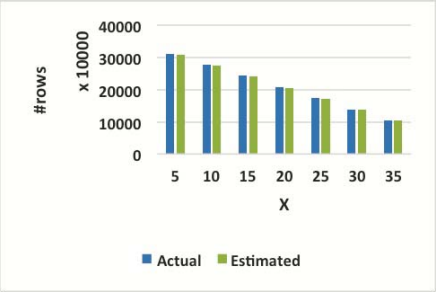


Figure 10: Selectivity Estimation for Q2

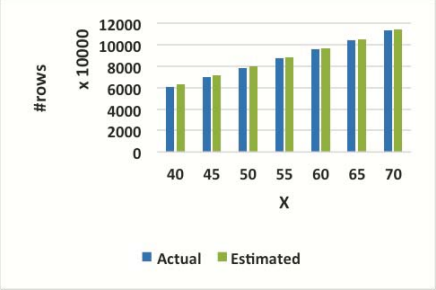


Figure 11: Selectivity Estimation for Q3

TABLE 5: MAPE for Selectivity Estimation

Query	Mean Absolute Percentage Error
Q1	1.565298873
Q2	0.872011458
Q3	1.568515777

6.3 Memory Estimation

In order to evaluate the memory consumption, we implement a monitoring program to monitor each Impala Daemon. When the program starts, it will connect the Impala StateStore to fetch the list of daemons. After the query is executed, monitor starts polling every Impala Daemon per second, and maintains the maximum of memory usage across all nodes. When the query is completed, the monitor will report the maximum memory usage.

To prevent from waiting for a large result set streaming back, we use a limit clause in the SQL. For example, in TABLE 6, we use "LIMIT 10" to fetch the first 10 rows of result.

As illustrated in Figure 12, if the scanning range is small, the data may not occupy all of the buffers. It makes our estimation in those cases slightly higher than actual memory usage. From this experiment result, the MAPE we get is 5.766993529%.

TABLE 6: Query for Memory Estimation

#	Query
Q4	SELECT * FROM transactions JOIN books ON (transactions.book_id = books.id AND books.price < X) LIMIT 10

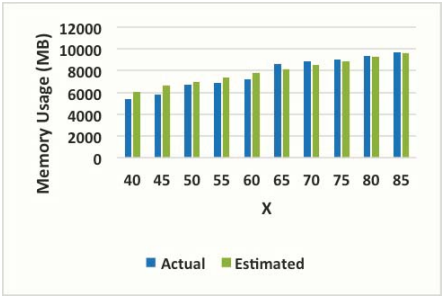


Figure 12: Memory Estimation

6.4 Performance Evaluation

We evaluate the performance of fusion SQL engine system and compare it to that only with Hive or with Impala. We set the memory limitation of Impala and maximum heap size of Hive to 8GB. Hive, Impala and the fusion SQL engine (FSE) will execute all the SQL queries in TABLE 7 for comparison.

Q5 is a normal scanning and aggregation query; Q6 is similar to Q5 with an additional filter predicate; Q7 is a sorting and aggregation query; Q8 causes two large tables join that does not fit in memory. Q9 is a complex query with sub query, sorting, join, aggregation operations.

In Figure 13, for most of the cases (Q5, Q6, Q7, and Q9), the query execution time in FSE and Impala is quite shorter than Hive. However, for the case of Q8, Impala cannot execute it because of the memory size limitation. It will fail at 34.437 seconds. In FSE, it chooses the Hive as the SQL engine to execute the query. Therefore, the performances of our proposed FSE and Hive are similar. The overhead of FSE is under 1 second; it is caused by the simple computation and the communication between Client and Selector.

The speedup is described in TABLE 8. In aggregation operations (Q5 and Q6), we get about 2.53x-5.18x better than Hive, and we get about 3.82x in the join operations (Q9). It shows that the fusion SQL engine system can improve the performance in most of the queries. Our fusion SQL engine can correctly select the best-fit SQL engine (Hive or Impala) to execute a query for a given dataset.

TABLE 7: Queries for Performance Evaluation

#	Query
Q5	SELECT COUNT(*) FROM transactions
Q6	SELECT COUNT(*) FROM customers WHERE name = 'Harrison SMITH'
Q7	SELECT category, count(*) cnt FROM books GROUP BY category ORDER BY cnt DESC LIMIT 10
Q8	SELECT * FROM transactions JOIN books ON (transactions.book_id = books.id AND books.price < 80) LIMIT 10
Q9	SELECT tmp.book_category, ROUND(tmp.revenue, 2) AS revenue FROM (SELECT books.category AS book_category, SUM(books.price * transactions.quantity) AS revenue FROM books JOIN transactions ON (transactions.book_id = books.id AND books.price < 80) GROUP BY books.category) tmp ORDER BY revenue DESC LIMIT 10

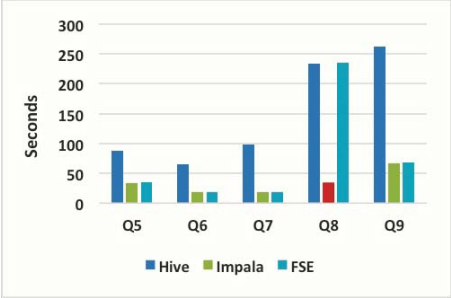


Figure 13: Performance Evaluation

TABLE 8: Speedup of FSE Based on Hive and Impala

#	FSE vs. Hive	FSE vs. Impala
Q5	2.53	0.98
Q6	3.50	0.96
Q7	5.18	0.94
Q8	1.00	N/A
Q9	3.82	0.98

7 Conclusions and Future Work

In this paper, we propose a fusion SQL engine (FSE) system on Hadoop framework. We can correctly select the best-fit SQL engine (Impala or Hive) to execute a query for a given dataset and therefore get the best performance. We also implement a failover mechanism. In the future, we will try to support partition table, add sampling based estimation.

Acknowledgement

The work of this paper is partially supported by Ministry of Science and Technology under contract MOST 103-2221-E-007 -063 -.

Reference

[1] *Apache Hadoop*. http://hadoop.apache.org/
[2] *Impala*. http://impala.io/
[3] *Impala Performance Testing and Query Optimization - Amazon Elastic MapReduce*. http://docs.aws.amazon.com/ElasticMapReduce/latest/DeveloperGuide/impala-optimization.html
[4] *Introducing Cost Based Optimizer to Apache Hive*. Available: https://cwiki.apache.org/confluence/download/attachments/27362075/CBO-2.pdf
[5] A. Abouzeid, K. Bajda-Pawlikowski, D. Abadi, A. Silberschatz, and A. Rasin, "HadoopDB: an architectural hybrid of MapReduce and DBMS technologies for analytical workloads," *Proceedings of the VLDB Endowment*, vol. 2, pp. 922-933, 2009.
[6] M. M. Astrahan, M. W. Blasgen, D. D. Chamberlin, K. P. Eswaran, J. Gray, P. P. Griffiths, *et al.*, "System R: relational approach to database management," *ACM Transactions on Database Systems (TODS)*, vol. 1, pp. 97-137, 1976.
[7] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, pp. 107-113, 2008.
[8] A. Ganapathi, Y. Chen, A. Fox, R. Katz, and D. Patterson, "Statistics-driven workload modeling for the cloud," in *Data Engineering Workshops (ICDEW), 2010 IEEE 26th International Conference on*, 2010, pp. 87-92.
[9] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The Google file system," in *ACM SIGOPS Operating Systems Review*, 2003, pp. 29-43.
[10] A. Gruenheid, E. Omiecinski, and L. Mark, "Query optimization using column statistics in hive," in *Proceedings of the 15th Symposium on International Database Engineering & Applications*, 2011, pp. 97-105.
[11] A. Lupher, "Shark: SQL and Analytics with Cost-Based Query Optimization on Coarse-Grained Distributed Memory," 2014.
[12] S. Melnik, A. Gubarev, J. J. Long, G. Romer, S. Shivakumar, M. Tolton, *et al.*, "Dremel: interactive analysis of web-scale datasets," *Proceedings of the VLDB Endowment*, vol. 3, pp. 330-339, 2010.
[13] A. Okcan and M. Riedewald, "Processing theta-joins using MapReduce," in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, 2011, pp. 949-960.
[14] G. Piatetsky-Shapiro and C. Connell, "Accurate estimation of the number of tuples satisfying a condition," in *ACM SIGMOD Record*, 1984, pp. 256-276.
[15] P. G. Selinger, M. M. Astrahan, D. D. Chamberlin, R. A. Lorie, and T. G. Price, "Access path selection in a relational database management system," in *Proceedings of the 1979 ACM SIGMOD international conference on Management of data*, 1979, pp. 23-34.
[16] Y.-J. Shi, X.-F. Meng, F. Wang, and Y.-T. Gan, "HEDC++: An Extended Histogram Estimator for Data in the Cloud," *Journal of Computer Science and Technology*, vol. 28, pp. 973-988, 2013.
[17] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," in *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on*, 2010, pp. 1-10.
[18] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, N. Zhang, *et al.*, "Hive-a petabyte scale data warehouse using hadoop," in *Data Engineering (ICDE), 2010 IEEE 26th International Conference on*, 2010, pp. 996-1005.
[19] R. S. Xin, J. Rosen, M. Zaharia, M. J. Franklin, S. Shenker, and I. Stoica, "Shark: SQL and rich analytics at scale," in *Proceedings of the 2013 international conference on Management of data*, 2013, pp. 13-24.
[20] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: cluster computing with working sets," in *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, 2010, pp. 10-10.

From in-disk to in-memory big data with Hadoop: Performance experiments with nucleotide sequence data

A. Radenski[1], L. Ehwerhemuepha[1], and K. Anderson[1]
[1]Schmid College of Science and Technology, Chapman University, Orange, California, U.S.A.

Abstract. *Apache's Hadoop, the de facto standard big data business analytics platform, has been increasingly used for big data projects in the sciences in general and in bioinformatics in particular. While the numerous strengths of Hadoop have been widely recognized, its deficiencies have been in the focus of constructive criticism. In particular, the inadequate run time efficiency of the original Hadoop MapReduce in-disk engine has driven an on-going transition to the more efficient Spark in-memory engine. It has been acknowledged that Spark has a pronounced efficiency edge over MapReduce; at the same time, strict performance comparisons and analysis are scarce. To help fill the relative void, we experimented with codon count algorithms on nucleotide sequence data. To do so, we measured the performance of a Spark codon count algorithm on the Amazon cloud platform and compared it to our earlier MapReduce algorithms: a basic codon count algorithm and an optimized "local in-memory aggregation" (or simply "local aggregation") algorithm. As expected, our experiments confirmed that in-memory codon count with Spark is much faster (about 15 times) than basic in-disk codon count with MapReduce. Surprisingly, however, in-memory codon-count with Spark remains about two times slower than optimized "local aggregation" codon count with MapReduce. This shows that properly optimized big data analysis with MapReduce can be faster than analysis with Spark, while working reliably with larger data sets that do not fit in memory. Our results can be beneficial to researchers and practitioners who need to choose a suitable big data execution model for their current needs.*

Keywords: Hadoop, MapReduce, Spark, codons, performance

1 Introduction

Big data is an informal term used to refer to data sets that cannot be stored and processed with widespread, off the shelf hardware and software systems. Big data has at least one of the following attributes: large volume, high velocity (very fast data) and significant variety (largely heterogeneous data) [17, 7, 14]. While there are no specific boundary delineating big data, the size of big data sets usually range from terabytes to exabytes. The size of data exchanged through telecommunication network in 1986, 1993, 2000 and 2007 were 281 petabytes, 471 petabytes, 2.2 exabytes and 65 exabytes respectively [37]. However, in 2012, 2.5 exabytes of data were created daily (this is more than all data generated in 2000) and doubling every 40 months [7].

The interest in big data lies in the opportunity to reveal hidden, non-obvious knowledge that can be derived through data mining, statistical analysis, machine learning and other suitable methods in business, science and engineering, healthcare, and virtually any realm of human activity.

Although various big data tools and platforms have been developed over the years, we chose to focus on the Apache Hadoop Ecosystem because it is currently the most widely known and used platform for storage and analysis of big data.

The origins of Hadoop can be traced back to the early 2000s with the development of a MapReduce engine and distributed file system for the Nutch web crawler. Nutch grew into Hadoop which in 2008 was elevated to a top-level Apache project [34, 1].

Historically, Hadoop was built for batch processing of large textual data across multiple commodity servers. It grew out of the need to process, in fault-tolerant manner, voluminous text data across commodity hardware in such a way that computation is moved to the data and I/O latency from moving large chunks of data is consequently averted. Since its development, Hadoop has been considered a low cost, scalable, flexible, and fault-tolerant alternative for batch processing large data sets. By its original design, Hadoop was intended to address mainly the "large volume" aspect of big data.

We refer to MapReduce as *in-disk* engine because it involves the file system in all communications, including those on the same node. While this provides fault tolerance and scalability, it is detrimental to performance. The performance deficiency of MapReduce has stimulated the development of Spark, a faster alternative to MapReduce [38, 28]. We refer to the Spark as *in-memory* engine because it uses, in contrast to MapReduce, as much as possible the entire available memory for data storage and communication.

The big data community has accepted that Spark has a pronounced efficiency edge over Hadoop. At the same time, strict performance comparisons and analysis are scarce. To provide additional systematic insight, we experimented with codon count algorithms on nucleotide sequence data. In particular, we measured the execution times of an in-memory (with Spark) algorithm on the Amazon cloud platform and compared them to the execution times of a basic and an optimized in-disk (with MapReduce) algorithms.

In the rest of this paper, we review the Hadoop ecosystem, including its in-disk and in-memory aspects, then describe our

experiments and compare the performance of Spark to the performance of MapReduce.

2 The Hadoop ecosystem

Since 2008, Hadoop and associated Apache projects have grown steadily to what is now referred to as the Hadoop Ecosystem. The Hadoop ecosystem comprises core projects and a number of related projects that can run on top or alongside of the core.

2.1 Core projects

The Hadoop core consists of the Hadoop Distributed File System (HDFS), the Hadoop MapReduce in-disk engine (MR), the Hadoop Common, and the YARN resource manager.

The *HDFS* is capable of storing very large data sets reliably and in a fault-tolerant way on potentially unreliable clusters of commodity servers. The HDFS design presumes that hardware failure is a norm; it also presumes that HDFS-based applications perform batch processing of large data sets, are write-once-read-many applications, are portable across heterogeneous commodity hardware, and are such that moving computation is cheaper than moving data [2]. The HDFS is not well suited for low-latency data access, data sets made of many small files, scenarios with multiple writers and arbitrary file modifications [19].

With *MR*, users specify serial map and reduce methods (one of each kind) that transform key-value records into new key-value records. The Hadoop MR implementation feeds input data to the mapper tasks and distributes intermediate key-value pairs to reducer tasks for final processing and output. In that, all intermediate records with the same key are distributed to the same reducer [24]. All communication in Hadoop MR goes via the file system.

The Hadoop *Common* contains libraries and utilities used by other Hadoop projects, while *YARN* is a resource negotiator that controls resource allocation within a cluster including application scheduling, also used by other projects. YARN was introduced to Apache Hadoop to decouple programming models from resource management and to delegate scheduling functions such as task fault tolerance to per-application components [33].

2.2 Related projects

The Hadoop ecosystem includes a variety of related projects that can interact with the Hadoop core, including the HDFS and the in-disk Hadoop MR engine.

Hive provides the ability to query and analyze large amount of historic (static) data in data warehouses by means of a SQL-like language, HiveQL. Hive translates higher-level user queries to lower-level Hadoop MR jobs, therefore freeing users from the relatively complex Hadoop MR Java API. Hive, however, is not a complete DBMS because record-level update, insert and delete operations cannot be performed directly by the underlying HDFS and MR. Hive is not well suited for rapidly changing data sets and is comparatively slower than traditional databases, partly because of the delay introduced by calling on and initiating MR jobs.

Pig is a high level programming language. The implementation translates Pig programs into Hadoop MR jobs.

HBase is a distributed, versioned, non-relational database modeled after, but not identical with Google's Bigtable [6, 10]. It can capture incremental data, such as user interactions in social networks and data produced by large cluster health monitors for example.

Presto is a distributed SQL engine for interactive big data analysis spanning from giga- to petabytes [23]. Presto accepts connections to different data sources (such as Hive, HBase, Cassandra, Scribe, relational databases, and proprietary data stores such as Amazon S3). Presto is optimized for ad-hoc analysis at interactive speed, supporting "standard ANSI SQL, including complex queries, aggregations, joins, and window functions" [4].

Mahout is a big data library of classification and recommendation algorithms. Originally implemented in Hadoop MR, Mahout is highly scalable and is able to support distributed processing of large data sets across commodity clusters.

Further examples include *Cassandra* (developed by Facebook in 2008 as an offshoot of BigTable), *Voldemort* (distributed key-value store created by LinkedIn in 2009), *Tajo* (SQL query engine for Hadoop created in South Korea in 2010), *Kafka* (data ingest framework originally developed by LinkedIn and open sourced in early 2011), *Storm* (stream computing framework released by Twitter in 2011), and *Impala* (SQL query engine created by Cloudera in 2012).

In recent years, *Spark* has gained popularity as an in-memory implementation of the map-reduce parallel model and is now emerging as a faster substitute for the original Hadoop MR in-disk engine. Because of Spark's growing importance, we discuss it separately in a later section.

2.3 Deploying and running Hadoop

Hadoop can be deployed on traditional on-site clusters as well on public, private, and hybrid clouds. It can run on virtual machines where it is known to perform marginally slower than the physical machines [13].

Several companies provide Hadoop in public cloud services at a cost to users. Three notable examples are Amazon (Amazon Web Services), Microsoft (Microsoft Azure), and Google (Google App Engine). We have chosen to work with the Amazon Web Services (AWS) which is the largest cloud computing platform in the world and which provides AWS usage grants to universities for research and teaching.

Amazon Elastic MapReduce (Amazon EMR) is a web service that makes it easy to quickly and cost-effectively process vast amounts of data. Amazon EMR uses Hadoop to process data across clusters of desired (by users) sizes. Launching an Amazon EMR is a high-level task with node provisioning, cluster setup, Hadoop configuration and cluster tuning all abstracted from the user; this abstraction allows first time users ease of use. EMR is reliable and provides fault

tolerance through automatic monitoring of the cluster to handle failed or failing nodes. It is "elastic" because any number of compute nodes can be provisioned and because it is easy to scale up or down and the user has complete control (such as root access) over the cluster. For security, EMR automatically configures firewall settings that control network access in a logically isolated user-defined network. A beneficial recent addition to AWS is Hue, a web-based graphical user interface for interactive access to AWS clusters (including EMR and HDFS) and the S3 cloud storage.

Cloudera provides an easy to install and configure pre-packaged distribution of Apache Hadoop, enhanced with custom Cloudera components. Other popular distributions include Apache Hadoop and MapR.

2.4 Hadoop deficiencies

There are, of course, drawbacks inherent in the original Hadoop architecture. Originally, Hadoop was specifically designed to run in a fault-tolerant manner long non-iterative batch jobs over large sets of static text data. Jobs over large numbers of small datasets can be inefficient. Interactive jobs and jobs that require data updates can be quite inefficient, too, if at all possible. The Hadoop MR engine employs the file system in all communications which can be detrimental to efficiency. These and other Hadoop difficulties are being addressed with new additions to the Hadoop ecosystem, most notably the Spark in-memory engine.

3 In-memory big data with Spark

Spark is a big data framework developed to take advantage of in-memory computation. Apart from increased speed, Spark provides support for cyclical data flow data model applications, thus eliminating another weakness of MapReduce [38].

Spark is a fast, general purpose engine for large-scale data processing that employs Resilient Distributed Datasets (RDD) and a distributed memory abstraction for in-memory computation on large clusters [28, 39]. Spark can be deployed as either a standalone application or on top of Hadoop. While Spark aims to fully utilize available memory, it also has the capacity to perform in-disk processing with larger data sets that do not fit entirely in available memory. Spark provides fault-tolerance through techniques that permit the restoration of RDDs upon node failure. Differences between MR and Spark are outlined in [9].

A Spark application involves the following principal components:

- Driver program written in Python, Java, or Scala.
- SparkContext object which is created within the driver program and coordinates the Spark processes running on the cluster. SparkContext connects to a supported cluster manager (such as YARN, Mesos, and Spark's own standalone cluster manager) and distributes tasks across worker nodes.
- Executor processes (referred to as executers) that carry out application-specific computations on the worker nodes under SparkContext.

The parallel computing primitives available in Spark include reduce, collect and foreach operations. Shared variables called broadcasters and accumulators are available to help with map, filter and reduce operations.

Spark is flexible framework as it can be run on Hadoop, Mesos, in the cloud or standalone, and can process a variety of data sources such as the HDFS, Cassandra, HBase and Amazon. Several organizations such as UC Berkeley AMPLab, Amazon, IBM Almaden and NASA JPL use Spark for building applications for large scale analytics and interactive exploration of large data [28].

Spark comes with higher level extensions for big data analytics, such as the SQL-like query extension, the MLlib machine learning extension, and the GraphX graph processing extension [28]. Most notably, the Spark Streaming extension was developed to process discretized data streams enabling real-time data analysis [38].

Given that Spark can handle static or slowly changing data as well as fast stream data, while at the same time maintaining a speed-up advantage over MR [39, 29, and 38], it is not surprising that Spark is considered as a viable in-memory alternative to the in-disk MR implementation. Projects, originally implemented with the in-disk MR engine have been ported (Hive) or are now being ported (Mahout) onto the Spark in-memory engine.

4 Hadoop bioinformatics applications

Since MapReduce was implemented as a module of the open-source Apache Hadoop platform, it has found application not only in business analytics, but also in various scientific and engineering domains, such as sets and graphs; artificial intelligence, machine learning and data mining; bioinformatics; image and video; evolutionary computing; a-life modeling, statistics; and numerical mathematics, including PDE solvers [26]. In this section we review some bioinformatics applications of the Hadoop ecosystem, including MR and Spark.

SparkSeq is a general-purpose genomic tool built with Apache Spark for next-generation sequencing (NGS) [35]. It provides convenient methods for common tasks in bioinformatics and genomic studies such as sample, exon and position encoding using Ensembl gene annotation forma [36]. The RSparkSeq package is available to connect SparkSeq with R. SparkSeq accepts BAM and BED [3, 18, and 31] files for a variety of analysis tasks such as nucleotide and genomic coverage, number of short reads, and others.

Adam is a tool for large scale genomics analysis that consists of data formats, APIs and algorithms implemented on top of Spark [20]. For a particular configuration, a “50 fold decrease in time required to compute base substitution was achieved in comparison to using BCF tools on a local file system” [32].

Error correction of erroneous bases is an important first (preprocessing) step, to precede genome assembly and variant discovery in high-throughput NGS. Apache Spark has been used to provide a parallel algorithm for screening NGS sequence data for errors, ensuring faster processing that Hadoop MapReduce [8].

SeqHBase was built on Hadoop and HBase for applications in whole-genome sequencing (WGS) and whole-exome sequencing (WES) [11]. Earlier on, the SeqWare Query Engine was developed in 2010 to provide an easy way to make the U87MG genome available to both skill programmers and non-programmers alike. The SeqWare Query Engine uses Apache HBase as the backend database because of its robust querying abilities and auto-sharding of data across commodity cluster via Hadoop [5].

BioPig is a sequence analysis toolkit built on the Hadoop MR engine and the Pig programming language. It contains modules such as pigKmer for computing the frequencies of each kmer as a histogram; pigDuster for searching known sequences for near exact match; and other modules such as pigDereplicator [22].

SeqPig is a similar scalable tool for sequencing large data sets in Hadoop. It uses Apache Pig for automated parallelization across computing nodes [27].

Cloud BioLinux is a publicly accessible VM that can be deployed on the Amazon EC cloud and on Eucalyptus. Cloud BioLinux provides, by default, a range of pre-configured "command line and graphical software applications, including a full-featured desktop interface and over 135 bioinformatics packages for sequence alignment, clustering, assembly and phylogeny" [16]. Cloud BioLinux has been used to analyze protein disorder for whole organisms and for obtaining all possible single sequence variants in protein coding regions of the human genome [15].

FX is an RNA sequence analysis tool that can be installed on local Hadoop clusters and Amazon EC2 cloud. FX provides an enhanced mapping of short reads by using references made of known genes and their isoforms [12].

Hadoop MR has been used for distributed a-life simulation on the cloud [25].

Additional bioinformatics applications of the Hadoop ecosystem can be found in [30, 24].

5 Performance experiments on AWS, the Amazon Cloud Platform

5.1 Algorithms on nucleotide sequence data

Codons are triples over the four DNA nucleotides traditionally represented by the letters A, C, G and T. Given a dataset of DNA nucleotides, codon usage calculation is expected to produce the frequencies of codons in the set [24]. The calculated frequencies can then be studied with various statistical methods for a number of purposes, such as back-translation of protein sequences to their probable DNA sequences, identification of protein-coding regions of DNA, and identification of regions that probably do not encode a protein [21].

In earlier research, we have developed MR streaming algorithms that gather statistics on codon usage count (both single codons and codon pairs). For each of these tasks, we developed a basic non-optimized MR algorithm and a MR algorithm optimized with local in-mapper aggregation or simply local aggregation (LA) [19, 24]. Local aggregation is a technique that helps reduce the intermediate data volume between mapper and reducer tasks in MR. For local aggregation, the mapper uses an in-memory data structure to aggregate multiple intermediate counts and emit them at once, rather than emitting multiple trivial counts. Thus, the mapper is explicitly designed to perform part of the reducer's work.In this paper, we limit ourselves to single codon usage count. Pseudo-code for our basic MR algorithm is shown in Fig. 1, while Fig. 2 offers the pseudo-code for the optimized MR LA algorithm.

Then we evaluated the performance advantage of local aggregation by running our basic MR and MR LA algorithms on Amazon's EMR cloud over sequence data from a tuberculosis database and by measuring the algorithms' execution times [24].

```
1: class Mapper
2:    method Map ()
3:       for line ∈ stdin do
4:          codn-list = Parse (line)
5:          for each codon ∈ codon-list do
6:             Emit(codon, count=1)

1: class Reducer
2:    method Reduce ()
3:       previous ← None
4:       for line ∈ stdin do
5:          codon, count = Parse (line)
6:          if codon ≠ previous then
7:          if previous ≠ None then
8:                Emit(codon, count-total)
9:             previous = codon; count-total = 0
10:         Increment (count-total, count)
11:      if codon ≠ None then Emit(codon, count-total)
```

Figure 1: Pseudo-code for the basic codon count algorithm in MR streaming. The mapper emits an intermediate key-value pair for each codon occurrence; the reducer sums up all counts for each individual codon using the fact that keys (i.e. codons in this case) are supplied in sorted order.

In this paper, we revisit the codon usage count problem and run a basic Spark implementation (Fig. 3) on Amazon EMR clusters that are identically configured as in our previous experiments. We have also repeated representative set of our previous experiments to confirm that current cluster performance remains the same. This approach permitted us to compare Spark and MR performance by accumulating performance data for Spark and reusing previously accumulated data for MR.

```
1: class Mapper
2:    method Map ()
3:       codon-hash = ∅
4:       for line ∈ stdin do
5:          codon-list = Parse(line)
6:          for each codon ∈ codon-list do
7:             Increment (codon-hash[codon], 1)
8:       for codon ∈codon-hash do
9:          Emit(codon, count = codon-hash[codon])
```

Figure 2: Pseudo-code for the mapper part of the LA codon count algorithm in MR streaming. The mapper uses an in-memory data structure to accumulate partial counts, before finally emitting all of them. The reducer is the same as in Figure 1.

```
1: method getCodons (nucleotide_sequence):
2: codon_list = []
3: for each codon ϵ nucleotide do
4:    codon_list.append (codon)
5: return codon_list

6: file <- SparkContext.textFile ("path_to_data_on_S3")
7: codons <- file.flatMap (for line ϵ file: getCodons(line))
8: codon_count_1s <- codons.map (lambda codon: (codon, 1))

9: results <- codon_count_1s.reduceByKey (lambda x, y: x+y)
10: results.saveAsTextFile ("path_to_output_storage_on_S3")
```

Figure 3: Pseudo-code for the Spark codon count algorithm.

5.2 Performance experiments with nucleotide sequence data on the Amazon cloud platform

We implemented a Spark codon count program (Fig. 3) in Python 2.7.9. To assess performance, we ran our Spark implementation of codon count algorithm as a bootstrap program on the Elastic MR cloud with Hadoop 2.4.0. In all experiments, current with Spark and previous with MR, we used four m1.large AWS instances – a master instance and three core instances. This permitted us to compare performance results without rerunning earlier time-consuming MR experiments. Likewise, we used exactly the same gene sequence data in both experiments, replicating the data to scale from 1GB to 100GB of sequence data.

We uploaded the collection of datasets onto an AWS's S3 cloud. After provisioning a cluster, we SSH-ed into the cluster's master node and submitted our Spark application by using "spark-submit", setting "--master" to "yarn-client" and "--num-executors" to "6". We chose six executors after performing several experiments on 25GB of data to determine the optimal number of executors, performance wise. As in the experimental setup for our earlier MR jobs, we ran the Spark application twice and calculated the average execution time. All performance results are presented in Table 1 and visualized in Fig. 4 and Fig. 5.

As previously known [24], our optimized LA algorithm runs 16 to 34 times faster than our basic MR algorithm (Table 1). As expected, our Spark algorithm runs (about 15 times) faster than our basic MR algorithm. Somewhat surprisingly, our optimized LA MR algorithm performs two times faster than our Spark implementation, at least for the particular datasets used in these experiments (Table 1). We attribute the performance advantage of MR over Spark to the custom in-memory local aggregation optimization.

Table 1: Performance data (measured in minutes on AWS's EMR) for the (i) basic MapReduce (MR), (ii) MapReduce with local aggregation (MR LA), and (iii) Spark algorithms for codon count, plus speed gains of MR LA against Spark and MR, and of Spark against MR.

Elapsed time (in minutes)												
Data size (GB	Basic MR algorithm			MR LA algorithm			Spark algorithm			MR / MR LA	MR / Spark	Spark / MR LA
	Min	Max	Avg	Min	Max	Avg	Min	Max	Avg			
1	31	32	31.5	2	2	2.0	2.3	2.3	2.3	16	14	1.1
4	106	120	113.0	4.5	5	4.75	7.9	8.0	8.0	24	14	1.6
9	263	270	266.5	8	9	8.5	17.5	17.5	17.5	31	15	2.0
16	465	482	473.5	15	16	15.5	33.8	34.3	34.1	31	14	2.2
25	720	780	750.0	21	23	22.0	46.3	47.5	46.9	34	16	2.1
36	-	-	-	34	34.5	34.25	66.3	73.4	69.9	-	-	2.0
49	-	-	-	40	44	42.0	91.8	102.3	97.1	-	-	2.3
64	-	-	-	54	59	56.5	119.8	119.8	119.8	-	-	2.1
81	-	-	-	71	74	72.5	147.8	148.0	147.9	-	-	2.0
100	-	-	-	85	86	85.5	182.6	227.6	205.1	-	-	2.4

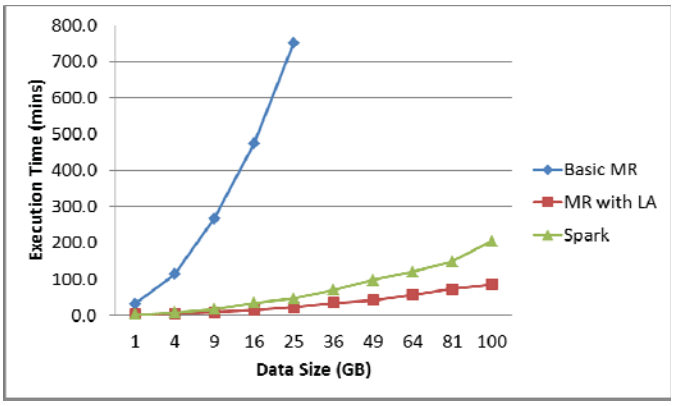


Figure 4: Execution times of the basic MR, the LA MR, and Spark algorithms for codon count.

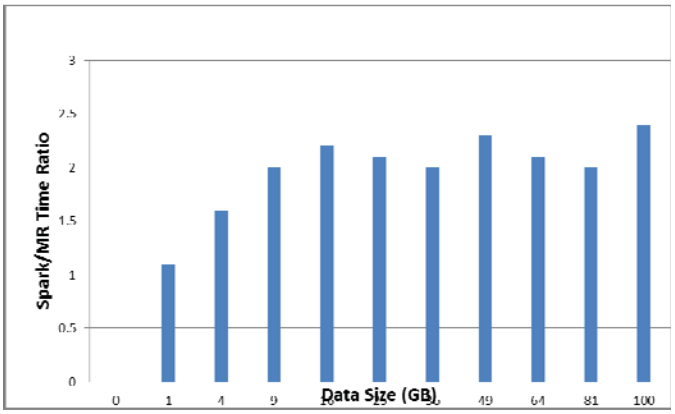


Figure 5: Performance advantage of the LA MR algorithm for codon count over the Spark algorithm.

6 Conclusions

In this paper, we analyzed the state of the art of the Hadoop big data echo-system and focused on the current trend from in-disk to in-memory big data processing. We reviewed and experimented with Spark, the emerging in-memory alternative to the classic Hadoop MapReduce framework.

To compare the performance of Spark and Hadoop MR, we executed, in Amazon's EMR cloud, Spark and Hadoop MR algorithms over large nucleotide sequence data. Specifically, we showed that optimized (with local aggregation) MR algorithm for simple codon analysis can be twice as fast as corresponding Spark algorithm; to the best of our knowledge, we the first to demonstrate the potential performance edge of MR due to custom LA. As expected, the performance of non-optimized MR lags behind Spark. We therefore suggest that available optimization techniques be considered for existing Hadoop MR applications before making a decision to re-implement them in Spark for performance gains.

We believe that this research can be beneficial to scholars and practitioners who need to choose a suitable big data execution model for their current needs.

7 References

1. Apache Hadoop, http://Hadoop.apache.org/ (Retrieved May 14, 2015).
2. Borthakur, D. (2007). The Hadoop distributed file system: Architecture and design. The Apache Software Foundation, (2007).
3. Broad Institute, https://www.broadinstitute.org/ (Retrieved May 14, 2015).
4. Chan L., Presto: Interacting with petabytes of data at Facebook, https://www.facebook.com/notes/facebook-engineering/presto-interacting-with-petabytes-of-data-at-facebook/10151786197628920 (Retrieved May 14, 2015).
5. D. O'Connor, B. Merriman, S. Nelson. SeqWare Query Engine: storing and searching sequence data in the cloud. Bmc Bioinformatics, 11(Suppl 12), S2, (2010).
6. George, L. (2011). HBase: the definitive guide. O'Reilly Media, Inc. (2011).
7. Gerhardt, B., Griffin, K., Klemann, R. (2012). Unlocking Value in the Fragmented World of Big Data Analytics. How Information Infomediaries Will Create a New Data Ecosystem, Cisco Internet Business Solutions Group (IBSG), 2013-2017 (2012).
8. Gong, Y. Parallel Algorithms for Screening NGS Data Using Spark (2014).
9. Gopalani, S., Arora R. Comparing Apache Spark and Map Reduce with Performance Analysis using K-Means. International Journal of Computer Applications (0975 – 8887), Volume 113 – No. 1 (2015).
10. HBase, A. A Distributed Database for Large Datasets. The Apache Software Foundation, http://hbase.apache.org (Retrieved May 14, 2015).
11. He, M., Person, T. Hebbring, S., Heinzen, E., Ye, Z., Schrodi, S., Wang, K.. SeqHBase: a big data toolset for family based sequencing data analysis. Journal of medical genetics, jmedgenet-2014 (2014).
12. Hong, D., Rhie, A., Park, S., Lee, J., Ju, Y., Kim, S., Seo, J. S. FX: an RNA-Seq analysis tool on the cloud. Bioinformatics, 28(5), 721-723 (2012).
13. Ibrahim, S., Jin, H., Lu, L., Qi, L., Wu, S., & Shi, X. Evaluating mapreduce on virtual machines: The hadoop case." In Cloud Computing, 519-528, Springer Berlin Heidelberg (2009).
14. Intel IT Center. Planning Guide: Getting Started with Hadoop. Steps IT Managers Can Take to Move Forward with Big Data Analytics (2012).
15. Kaján, L., Yachdav, G., Vicedo, E., Steinegger, M., Mirdita, M., Angermüller, C., Rost, B. (2013). Cloud prediction of protein structure and function with PredictProtein for Debian. BioMed research international (2013).

16. Krampis, K., Booth, T., Chapman, B., Tiwari, B., Bicak, M., Field, D., Nelson, K. E. Cloud BioLinux: pre-configured and on-demand bioinformatics computing for the genomics community. BMC bioinformatics, 13(1), 42 (2012).
17. Laney, D. 3D data management: Controlling data volume, velocity and variety. META Group Research Note, 6 (2001).
18. Li H., Handsaker B., Wysoker A., Fennell T., Ruan J., Homer N., Marth G., Abecasis G., Durbin R. The Sequence alignment/map (SAM) format and SAMtools. Bioinformatics, 25, 2078-9 (2009).
19. Lin, J., Dyer, C. Data-intensive text processing with MapReduce. Synthesis Lectures on Human Language Technologies, 3(1), 1-177 (2010).
20. Massie, M., Nothaft, F., Hartl, C., Kozanitis, C., Schumacher, A., Joseph, A., Patterson, D. Adam: Genomics formats and processing patterns for cloud scale computing. EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2013-207 (2013).
21. McInerney, J. GCUA: general codon usage analysis, Bioinformatics 14(4), 372-373 (1998).
22. Nordberg, H., Bhatia, K., Wang, K., Wang, Z. BioPig: a Hadoop-based analytic toolkit for large-scale sequence data. Bioinformatics, 29(23), 3014-3019 (2013).
23. Presto, https://prestodb.io/ (Retrieved May 14, 2015).
24. Radenski, A., Ehwerhemuepha, L. Speeding-up codon analysis on the cloud with local MapReduce aggregation, Information Sciences, Elsevier, 263, 175-185 (2014).
25. Radenski, A. Using MapReduce streaming for distributed life simulation on the cloud, Advances in Artificial Life, ECAL 2013, MIT Press, 284-291 (2013).
26. Radenski, A. Big data, high-performance computing, and MapReduce. Proceedings of the 15th International Conference on Computer Systems and Technologies (CompSysTech '14), ACM, New York, NY, USA, 13-24 (2014).
27. Schumacher, A., Pireddu, L., Niemenmaa, M., Kallio, A., Korpelainen, E., Zanetti, G., Heljanko, K. SeqPig: simple and scalable scripting for large sequencing data sets in Hadoop. Bioinformatics, 30(1), 119-120 (2014).
28. Spark, Spark: Lightning-fast cluster computing, https://spark.apache.org/ (Retrieved May 14, 2015).
29. Tabaa, Y., Medouri, A., & Tetouan, M. Towards a next generation of scientific computing in the cloud. International Journal of Computer Science (2012).
30. Taylor, R. An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics. BMC bioinformatics 11.Suppl 12 (2010): S1.
31. UCSC Genome Bioinformatics, http://genome.ucsc.edu/ (Retrieved May 14, 2015).
32. van Hagen, S., Schoots-van der Ploeg, J., Weistra, W., van Bochove, K. Evaluation of Spark and ADAM for large scale genomics data.
33. Vavilapalli, V. K., Murthy, A. C., Douglas, C., Agarwal, S., Konar, M., Evans, R., Baldeschwieler, E. Apache Hadoop yarn: Yet another resource negotiator. In Proceedings of the 4th annual Symposium on Cloud Computing, p. 5, ACM (2013).
34. White, T. Hadoop: The definitive guide. O'Reilly Media, Inc. (2012).
35. Wiewiórka, M., Messina, A., Pacholewska, A., Maffioletti, S., Gawrysiak, P., Okoniewski, M. SparkSeq: fast, scalable, cloud-ready tool for the interactive genomic data analysis with nucleotide precision. Bioinformatics, btu343 (2014).
36. Wiewiorka M. SparkSeq, https://bitbucket.org/mwiewiorka/sparkseq/wiki/Home (Retrieved May 14, 2015).
37. Wikipedia, Big data, http://en.wikipedia.org/wiki/Big_data (Retrieved May 14, 2015).
38. Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., Stoica, I. Spark: cluster computing with working sets. Proceedings of the 2nd USENIX conference on Hot topics in cloud computing, p. 10 (2010).
39. Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., Stoica, I. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation, p. 2 (2012).

SESSION

BIG DATA, DATA CENTERS AND CLOUD COMPUTING

Chair(s)

TBA

Improving Backup and Recovery of Modern Data Centers

Zhiwei Zhao, Haolun Yan, and Hussain Al-Asaad
Department of Electrical and Computer Engineering
University of California-Davis

Abstract

With the size of data sets becoming larger, the requirement of data backup and recovery is increasing significantly. In this paper, we first investigate the current backup techniques and analyze the advantages and drawbacks to determine the problems of these methods. Based on that, we established a new system to modify and optimize the process of the enterprise class data center, in order to make the transmission safer, faster, and more accurate. The system contains two parts: backup and recovery. The backup part is based on general network protocols and contains three layers to guarantee the data can be transmitted well. For the recovery part, the system takes an accurate shortest path method to eliminate the cost, which makes the process economical. Within the structure of this system, by using Matlab simulation, the optimal configurations of the backup network can be found. Meanwhile, the shortest path algorithm under the Java environment provides the lowest cost path to recover the data.

Key Words: Data Center, Network Structure, Backup, Recovery, Transmission Efficiency, Simulation.

I. Introduction and Relevant Work

Traditional data center backup technology contains plenty of methods such as bus, distributed, personal cloud, etc… In this section, we analyze these methods so that we can find out the characteristics of them and establish a theoretical system to overcome the drawbacks.

1. Bus: Oracle's data backup architecture is designed as a solution for providing network backups of heterogeneous clients, which is a kind of modified bus backup structure. Based on common implementations of existing customer environments, the target environment includes backup clients connected via a bus with 1 gigabit Ethernet network links [1].

A key objective of this architecture is to achieve competitive price/performance. Oracle's evaluation of the requirements established the number of backup targets that could be supported and provided an understanding of the system configuration and steady-state CPU utilization required to reach that performance level.

In this architecture, StorageTek Enterprise Backup software is used to provide disk-and tape-based data protection as well as archival and recovery management. The solution scales to support an enterprise network with numerous clients running a mix of Oracle Solaris, AIX, HP-UX, Linux, and Windows.

The whole structure includes backup clients, a management console, backup server, one or more storage nodes, and a wide variety of disk and tape devices that can be configured at the discretion of the system administrator in charge of the installation.

The configuration in Figure 1 serves a network environment with numerous backup clients running heterogeneous operating systems, all of which are configured to simultaneously back up their data to the Avamar solution. Although this architecture demonstrates many data backup methodologies, none of the backup devices relies on any other backup device for performance.

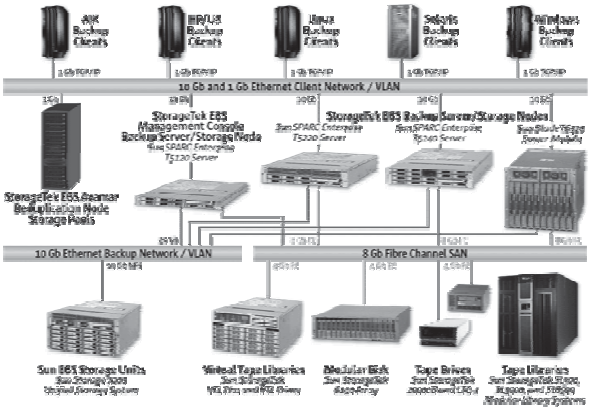


Figure 1. A network environment with backup [Figure imported from www.oracle.com].

In general, based on the bus structure, when transmitting the packets, the speed can be high enough to the bandwidth of the bus. Furthermore, considering

about the expanding, if clients are asking for the backup and recovery service, they just need the authority to link the devices to the bus. Bus architecture is fast and convenient.

However, the drawbacks are obvious. Bus is a kind of local serial communication architecture, which can be regarded as a broadcasting method. These architecture cannot be used to a long-distance transmit, but only be used in the local area network. Also the bandwidth is the bottleneck. As mentioned above, the transmission rate can reach to the bandwidth. When the data set is large enough, say 10 TB, the burden of the bus will be too heavy. Moreover, if adding plenty of devices into the bus, the error checking will be a big problem. Considering one device meets an error, and asks the server to transmit the packet again, the server will have to send that packet as a broadcasting way [2]. This is no doubt a waste of the network resource. Also, when recovering, one client asks the service means that the server needs to broadcast the data set even though the others do not need that. All in all, this method is expensive not only on the cost but the usage of resources.

2. Cloud: Cloud technology contains public cloud, private cloud, and mixed cloud. To personal customers, public cloud is easy to use. Like Google Drive, Dropbox, customers just need to update the files or data sets that they want to store to the cloud servers. In general, service providers provide public cloud mainly to the personal or small business customers. Since the public cloud cannot guarantee the safety, for its low-level encryption mechanism (in order to reduce the cost). Meanwhile, the capacity of storage is limited, so that people have to pay a lot for a larger capacity. Private cloud is used generally for the enterprise class data service. Figure 2 shows a campus private cloud data center.

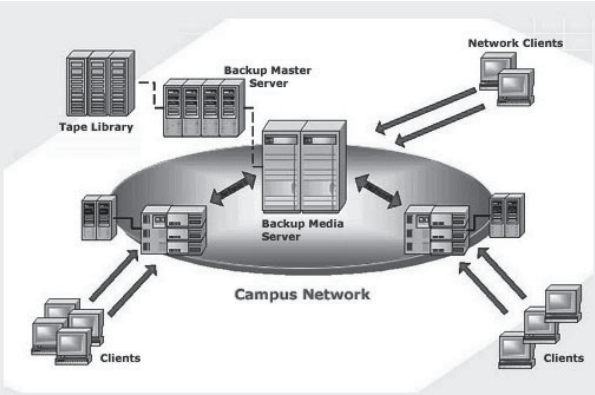


Figure 2. A campus private cloud data center [Figure imported from www.infotechsa.com].

Private cloud provides a relatively reliable and safe backup service. The network can be regarded as a local area network. Each client can reach and get the data directly from the private backup server. The server may prevent the access request from the outside links, unless it can get the authorization from the administrator. This mechanism is able to protect the data, and also the speed in the local area network can be guaranteed.

Apparently, there are drawbacks to the above method. Private cloud is too limited, for the LAN structure, and the strict access right. Since it is for the enterprise class customers, regardless of the cost, increasing the safety means the sacrifice the convenience. Considering about the recovery, if disaster happens, the server which is obviously a centralized structure can easily be destroyed. The worst case is that the outside can hardly access the data [3].

Mixed cloud is the combination of public and private cloud. In this case, the less important data can be stored in public clouds, and the important date is stored in the private ones. However, the drawbacks for enterprise class backup still exist.

3. Distributed: The modern data center is more like distributed structure. For the enterprise class data center, by implementing the distributed structure, big data sets can be split into several parts, and stored into different servers [4]. In order to guarantee the integrity of the data sets, it is necessary to add the redundant servers to store the same parts of data.

Every time, when a file is inserted into the backup sequence, it will be separated into blocks with specific headers. For safety, every block can be encrypted using convergent encryption, that is, the key used in the applied shared key cryptographic algorithm consists of the hash of the unencrypted file block [5].

This approach can keep the safety of data center. When disaster happens, the distributed structure can prevent the data set from the damage of a centralized data backup structure. Some of the server bases may be destroyed, while the redundant server bases still work and can recover the data sets. Considering about this mechanism, the separated data blocks can reduce the burden of the network transmission. Although the increasing number of server bases will make a higher cost, by reducing the network burden, it can be neutralized.

However, for a long-distance data backup, this structure has to rely on the stability of the network. Distributed structure just provides a basic idea for building the network, but not the consideration of the detailed configurations. Also, we still need to know how to neutralize the high cost of server bases and the reduction of the network burden.

II. Modified Backup/Recovery Structure

1. Backup Structure: According to the technologies mentioned above, we propose to design a new structure, which is based on the distributed one, and by adding more detailed methods, including long-distance data transmission, peer-to-peer (P2P) protocol, layered network architecture, etc… With these methods, we can overcome some of the drawbacks and make the enterprise class data center backup more reliable.

In general, the structure is designed as three layers. The first layer is regarded as the linkage between the different biggest data centers. In order to finish the long-distance data transmission, we use the fiber cable to guarantee the data quality and the transmission speed. Layer 2 contains plenty of nodes, which are distributed as the mesh topology. The original data set is divided into a number of pieces, which are transmitted to each node with a unique characteristic key value. Layer 3, which is under the layer 2, contains several traditional server rooms. Each of them stores the same part of the data set.

Figure 3 shows a simple structure of layer 1. Given the big data backup and the present data center architecture, each of the centers is a typical distributed structure, which is like the cloud structure. In each cloud, there are plenty of server bases, and a central server base, connected with each other in physical or virtual links. Considering a long-distance data set transmission for the enterprise class backup service, it is necessary to choose the fiber cable for its stability and high speed. In order to reduce the cost, we only use fiber cable in this layer.

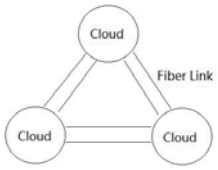


Figure 3. Network Topology- Layer 1 (Could Layer).

Layer 2 contains several nodes linked with the central data center as shown in Figure 4. This layer is under layer 1, which means that each cloud contains such a mesh network structure in it. In this layer, the original data sets will be split into pieces with a particular key value, in order to combine them into the original ones when recovering [2].

Layer 3 (shown in Figure 5) is under layer 2 obviously. After finishing the split process and each node has received one piece of data, and in order to keep the rational redundant data, the servers in this layer will store the same piece of the original one. Still considering about the transmission cost and speed, P2P protocol is a good method to not only eliminate the burden of the network, but also even accelerate the transmission process. The data node incoming part is like a server, based on P2P, and it only needs to transmit the whole data set once. Each of the servers under this layer will get some part of the data, and then they utilize a sharing mechanism to send the part they have already gotten, and receive the others that they have not had yet. This network does not request a huge of network resources. In contrast, it reduces the transmission amount of the expensive part, instead using a cheaper network to finish the process. Meanwhile, it guarantees the speed, since the P2P protocol theoretically can use the maximum bandwidth of the network [6].

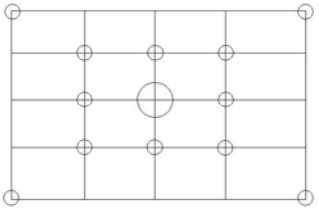


Figure 4. Network Topology-Layer 2 (Mesh Layer).

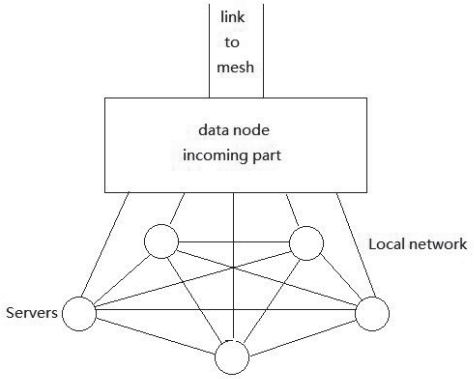


Figure 5. Network Topology-Layer 3 (Group Layer).

2. Recovery Structure: After finishing the backup part, the data center has been stored in different server bases. It means once the customers ask a data recovery service, they can choose one of the server bases to download the data.

Here the recovery structure mainly focuses on the cost of the process. Since there are multiple server bases, it is necessary to choose a lowest cost one to fetch the data back. A* algorithm, which is a modified Dijkstra Breadth-First-Search algorithm, is a good method to investigate the lowest cost path in a static network. However, we can modify this algorithm to figure out the lowest cost path even in a dynamic network. The description below is the introduction and Pseudo code of A* algorithm.

The key point to implement the A* is to set the admissible heuristics. Heuristic is regarded as the ideal cost from the current node to the destination. Once we record the cost of each link of the network in an ideal condition, it can be used as the heuristic to identify whether the path is the lowest cost one [7].

The formula $f(n) = g(n) + h(n)$ represents the strategy to find out the path. Here $g(n)$ means the cost from the initial node to the current node, and $h(n)$ means the admissible heuristic value based on the network's link cost. So, to find out the lowest cost path, the method needs to record the cost from the initial node, and add it to the heuristic value. By comparing all the values of $f(n)$ and choosing the smallest one, we can get the lowest cost path.

Below is the pseudo code of the algorithm.

```
Create two lists; //Open list saves all the created but not checked
nodes, Closed list records all the accessed nodes.
Compute the f(n) value of initial node;
Put initial node into Open list;
while(Open!=NULL)
{
    take out the node with the smallest f(n) value from the Open
list;
    if(Node n == target)
    break;
    for(each child node X from the current node n)
    {
        compute the f(n) value of X;
        if(X in Open list)
        if(f(n) of X is smaller than that of Open list)
        {
            set n as the parent of node X;
            upgrade the f(n) in the Open list; //get the f(n) of
        the shortest path
        }
        if(X in Close list)
            continue;
        if(X not in both)
        {
            set n as the parent of node X;
            compute the f(n) value of X;
            inset node X into Open list; //not sort yet
        }
    } //endfor
    insert node n into Closed list;
    sort the Open list nodes based on f(n); //Compare the value of
nodes in Open
    list, process from the smallest node.
}//endwhile(Open!=NULL)
Save the path;
```

Actually, the algorithm above is good for the static network. When dealing with the dynamic network, the cost from node to node may be changed. It is necessary to set an update frequency for the cost table. When using the A* algorithm to compute the lowest cost path, the child node's $f(n)$ can be updated in a specific frequency. Based on this structure, considering a situation, that one customer is asking a data recovery service. If there exist multiple server bases that can provide the service, and admissible heuristics of each base, each base can use A* algorithm to compute the cost from themselves to the target. Then by comparing the cost to choose the lowest cost one to provide the service.

III. Experiment and Simulation of the Backup/Recovery Structure

After the whole system is established, the next step is to do the simulations to find out a particular value for the number of nodes, error rate of transmission, accuracy of the network, etc... By combining every aspect, we can optimize the system into a relatively ideal state.

1. Simulation of backup: First, the necessary definitions and parameters should be clarified.

Definitions:

TSN = Total number of small stations
NG = Number of groups
TBW = Total bandwidth between two centers
TDS = Total data size
ERR = Error rate between nodes / GB
SBW = Station incoming bandwidth
NTR = Network traffic jam rate (specified by time)
MSS = Maximum station storage for backup
Assume total backup time is T
T = T1 + T2
T1 = Time of transmission from another Center
T2 = Time of local backup (peer to peer)
Assume all the backup is V2V which is easier to store, mark changes and transmit.
T1 > TDS/TBW
T2 is complicated, so we make more assumptions.

Assumptions:

MSS is at most 1/5 of the TDS, which means NG at least equals to 5.
NTR make the speed reduce to 10% of the transmission speed, when traffic jam is happening, which happens in 20% of the time.
Error rate between local nodes is low, which is 10^{-6} rate per GB, and only be used for calculating the accuracy during the P2P transmission and direct transmission. Calculate the re-transmission only in the direct mode.
Consider the error happening at most onetime per gigabytes transmission, since if it happens two or more times the probability could be low enough to be ignored.
Consider the miss rate inside the P2P transmission, but not a complex module. The re-transmission time caused by the miss rate can be ignored in such a simple module, in order to simplify the computing module.

Traffic Module:

Assume that each node normally transmits the data using the optimized speed, which may not be the maximum bandwidth, because in practice each node cannot use the whole link. When there is traffic jam that means the network is beginning to have more packets dropped and miss ordered, which due to the lower rate of transmission. So on the receiver sides, they are getting more error data and need more re-transmission. 10% of speed causes the error rate to become 10 times larger. The traffic jam is same to the whole grid network, which can be considered as the number of people who use the internet and in what time they use the network most. 20% is the assumption based on the statistic of the timing on people heavily using the internet. And we assume that each station has certain and equal speed in both downloading and uploading, which is easier to calculate the P2P network's time and accuracy.

Import the basic transmission time for custom P2P network:

$$T2 >= Max\{\frac{NF}{UTotal}, \frac{F}{D\min}\}$$

We denote the size of the file to be distributed (in bits) by *F* and the number of peers that want to obtain a copy of the file by *N*. The servers' total upload rate is *UTotal* and *Dmin* denote the download rate of the peer with the lowest download rate.

Now we are going to use this function into our specific situation. Call back all the assumptions and also try to develop the function in the worst case. Here make an assumption of the worst case. In general the worst should be for every Gigabyte it transmitted, there is a miss that causes re-transmission. And for the data integrity, we can take one Gigabyte as a block in our condition. When a miss happened, it means this block of data is corrupted. Thus we need to re-transmit the whole block, which may happen in P2P network very often. But in reality we cannot have a system running in such high miss rate, we shall make the worst case more realistic, such as make it 25% miss for worst case. We apply the above to timing/accuracy calculations.

In our case now, we can look into the function using 'F/Dmin' mode to complete the one we need.

$$F = \frac{TDS}{NG}$$

$$D(normal) = fixed - downloading - speed$$

Assume that the backup time is long since the data set is very huge so that it will take the 20% of traffic jam time and reduce the downloading speed heavily. We also assume that only one node will suffer from miss error at one time because it is a very small possibility that two stations both got the data miss.

Let's do this in a simple way. The worst case is that all the stations after all suffering from 25% of data miss in different time. Because the whole network is fair and equal, we can treat all the stations as one station with 25% of miss while others have no miss. The simulation then becomes simple. First we put the whole data set into pieces, and then take the 25% of miss to be the largest miss rate for the whole data set, which means it is the largest time of transmission. For the accuracy it is a different simulation, but still it is a function related to the group number.

$$D\min = D(normal) \times ((1 - NTR) \times (\frac{1}{2} \times 25\% + 0.75) + NTR/10 \times (\frac{0.25}{2} + 0.75))$$

$$T2 >= \frac{F}{D\min}$$

$$T2 >= \frac{TDS/NG}{D\min}$$

$$T >= TDS/(NG \times D\min) + TDS/TBW$$

Now we get the function of optimizing the transmission time. It is a function of number of groups. And we also need the minimum group number according to the disk size left for backup; this will give the lower limit of the function. There should also be an upper limit, but we will talk about that later.

Then it is necessary to discuss the accuracy in the transmission. Now the ERR is in use, and we shall consider most of our assumption in order to get a good function. In the P2P module we calculate the accuracy based on every single node in the network, which means we should consider all the nodes with ERR. Since the ERR defined is not with respect to time, we will also cover the whole time with the NTR, which still assumes that TDS is so large that will give enough time to get the NTR rate.

Simulation Process:

For a single node, we first figure out how large the data one node will receive. For uploading, since it is the same with downloading, which covers other nodes' downloading, there is no need to simulate it again. The data set's size should be *F*, which is the same in the timing function.

$$F - TDS/NG, unit - in - GB$$

$$ACC(Total) = ACC(Single)^{TSN}$$

$$ACC(Single) = ((1 - NTR) \times ERR + NTR \times ERR(traffic))^{F}$$

$$ACC(Total) - ((1 - NTR) \times ERR + NTR \times ERR(traffic))^{TDS \times TSN/NG}$$

This function also has the lower limit of minimum group number.

Now we have the two functions to do the whole optimization, we will run it in Matlab to see the result. Here the process graph in Figure 6 is to show how the backup works.

Now we have all the formula to write code for simulating them. For this part, since we are going to find the best group number for our condition, we will use the assumptions above and find out the best value from them. The result is supposed to turn out to be two separate lines. In order to compare the result, we shall combine them together to form a 3D graph and find out the best value of group numbers.

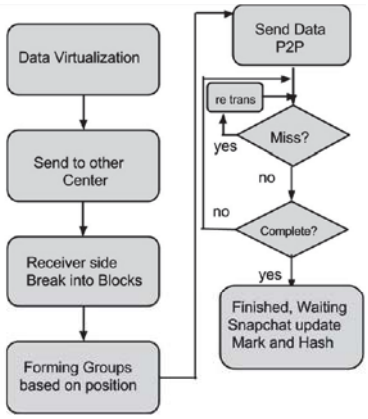


Figure 6. Process graph for backup.

Figure 7 shows the transmission accuracy, which means within particular number of servers, the accuracy tendency is increasing with augment of the group number. In this simulation, we assumed that there are 5000 servers, and divided them into groups. After 800 groups, we can find out that the accuracy increased exponentially, which means the performance of the network improved.

Figure 8 shows the transmission time. According to the tendency, we can also get that by increasing the number of groups the time is reduced. But we can find after 500 groups, the effect of NG is not that significant than before.

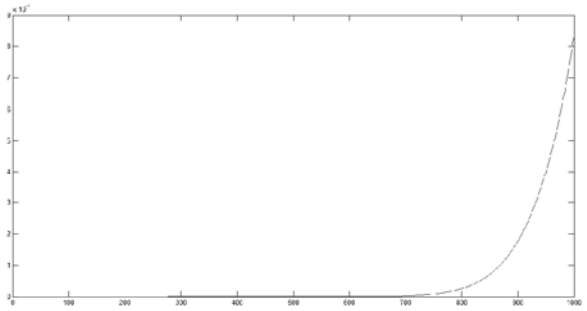


Figure 7. Transmission Accuracy.

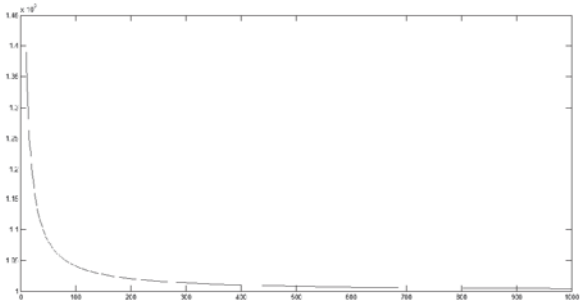


Figure 8. Transmission Time.

Combing time and accuracy, we get Figure 9 to show the general performance of the backup/recovery system. All in all, under the condition that redundant data can be guaranteed, the more group numbers, the better performance. So when implementing the system, we can obey this conclusion.

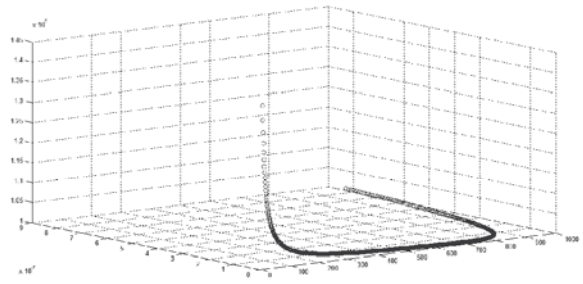


Figure 9. Transmission Time, Accuracy and Group Number.

2. Simulation of recovery: Now we will also simulate the recovery process. In this part we cannot do lots of assumptions to make it simpler, which will absolutely limit the result. We are using A* algorithm and cost map to simulate it. The best case would be using every node's uploading speed, and if they are equal to *U*, then the time for transmission would be $T >= F/U$, where F is the same as in the backup part.

$$F = \frac{TDS}{NG}$$

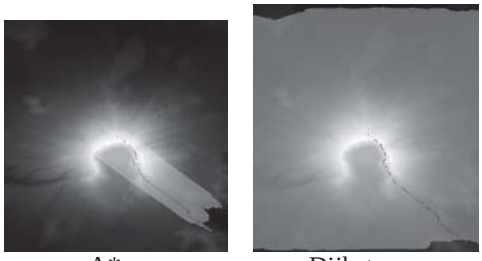
What we will simulate here is to try to get the best case, eliminate the cost and avoid the traffic. So A* is very important. Considering a real network grid, it contains lots of nodes with the ability to receive, transmit, acting like a router and more. Every node has its own cost table which update in real time to get the most recent network cost. From here we can acquire the real time cost in the network to calculate and find out the best route to the data center.

Here is the pseudo code for the whole process:

```
datacenter.requestdata(groups);
For( each groups){
        for(stations in group){
                get(cost table);
                calculate(cost);
                route=Astar(cost,nodes,grid);
                SendToCenter(groups.route);
        }
}
routes<routes> = datacenter.compare(groups.routes);
groups.senddata(datacenter, route.head,head.next);
while(!datareceivecomplete){
        do for each groups:
        get(recent cost table);
        calculate(cost);
        Astar(cost,newnode,grid);
        SendToCenter(groups.route);
        routes<routes>=datacenter.compare(groups.routes);
        groups.senddata(datacenter, route.head,head.next);
}
recoverydata = datacenter.reform(groupdatasets);
```

```
if(remotecenter.request){
        datacenter.send(remotecenter,recoverydata);
}
```

Here by running the A* and cost map, we can get the lowest cost path from one to another node, which is shown in Figure 10.



A* Dijkstra

Figure 10. A* and Dijkstra lowest cost paths.

By implementing the algorithm with Java, we load a preset cost map, and use A* algorithm to eliminate the number of nodes, which is necessary to be expanded. Considering about the real time characteristic, the fewer nodes expanded, the less time spent. Since the cost of each node is dynamic. By reducing the time of searching the lowest cost path, the accuracy to get that path can be much higher.

We can compare this algorithm with Dijkstra algorithm, which is also shown in Figure 10. The time taken of Dijkstra is almost 5 times than that of A*. It is obvious that the recovery strategy based on A* algorithm can have a better performance. As we mentioned above, since the network cost is dynamic, the less time spent means that the system would not update the cost table very often. From this angle, it can also improve the recovery efficiency.

IV. Conclusions

In this paper, we have mainly discussed a theoretical backup/recovery structure, which is regarded as a combination of cloud, distributed, broadcasting and P2P structures. Our structure absorbed the advantages of existing methods, and can overcome the drawbacks. It makes the enterprise-oriented backup/recovery service economic and efficient.

We can consider the security problem of the data transmission. Since the splitting of the data, each piece of the data contains a particular key value to guarantee the order, when recovery is requested. The strategy of recovery can be designed by the customers themselves. It means that the way to add the key value and how to recover the data in order is based on the key value. Each part of the strategy is unique, so that the security of data can be protected. In order to make it much safer, double-encryption can be considered. Under the strategy of key value, by encrypting the piece of data with the advanced algorithm, like AES 256 [8].

In our system, we designed three layers for the backup service, mainly based on the idea of distributed structure. We divided the data set into groups, and stored each of them. According to the simulation done in the paper, we can make a conclusion that under a specific number of servers, when the necessary redundant data can be guaranteed, the larger the number of groups, the better the network performance.

For the recovery part, by using the dynamic cost table, and A* shortest path algorithm, we can find out a lowest cost route in the dynamic network. Once one client needs to receive the particular part of data set, it can broadcast a particular value, so that each of the servers contains that data set will implement the dynamic A* algorithm to find the path. Then system will compare the paths and can choose the lowest cost one to transmit the data set.

As we know the data centers nowadays are based on virtual machines. Here we update the data center frequently by using the snapshot technique. The snapshot contains the changes of the data center but not the whole data [9]. In this way, the backup process can be more efficient.

In general, by combining the methods above, this backup/recovery system has the features of economic, efficiency, and anti-risk ability.

References

[1] J. S. Bozman, “Addressing virtualization and high availability needs with SUN Solaris cluster”, *IDC White Paper–Sponsored by SUN Microsystems*, 2009.

[2] M. G. Megerian, “Method and distributed database file system for implementing self-describing distributed file objects”, U.S. Patent No. 5768532, 1998.

[3] Y. Tan et al., “Sam: A semantic-aware multi-tiered source de-duplication framework for cloud backup,” *Proc. International Conference on Parallel Processing (ICPP)*, 2010, pp. 614 – 623.

[4] E. Murphy and N. Virk, “Distributed backup and versioning”, U.S. Patent No. 8769055, 2014.

[5] M. Teicher and T. Segalov, “Method and system for maintaining backup of portable storage devices”, U.S. Patent No. 7240219, 2007.

[6] R. Schollmeier, “A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications”, *Proc. IEEE International Conference on Peer-to-Peer Computing*, 2001, pp. 101-102.

[7] W. Zeng and R. L. Church, “Finding shortest paths on real road networks: the case for A*”, *International Journal of Geographical Information Science*, 23(4): 531-543, 2009.

[8] A. Biryukov and D. Khovratovich, “Related-key cryptanalysis of the full AES-192 and AES-256”, *Advances in Cryptology–ASIACRYPT*, 2009, pp. 1-18.

[9] B. Alpern et al., “The Jalapeno virtual machine”, *IBM Systems Journal*, Vol. 39, No. 1, pp. 211-238, January 2000.

On Applying Big Data and Cloud Computing for Quality Improvement in Higher Education

Shakeel Ahmed[1], **Hemant Kumar Mehta**[2]
[1]Computer Science Department, King Faisal University, Hofuf, AlHassa, Saudi Arabia
[2] Senior IEEE Member, Indore, India

Abstract - This paper presents a cloud and big data based approach for improving the quality of higher education. We developed a framework (CHEQEF) for analyzing the quality of higher education and further improving it. This framework uses the data available from the institutions under the study and feedbacks of various stakeholders including students, faculty members and industry delegates. The process starts with the analysis of examination results and later the stakeholder's feedback on the analysis of results. The framework focused on enhancement of syllabus and subjects in the various courses in the institutions under the study that ultimately results in the overall quality of the education in the higher educational institution. Moreover, this framework can also be adopted for school education with minor medication. The framework uses Hadoop and associated products for computations.

Keywords: Big Data; Cloud Computing; Higher Education; Quality; Optimization

1 Introduction

During the last two decades the evolution of distributed computing has changed the working of scientific and commercial applications. This progress has evolved several newer applications. The latest evolution of distributed computing is Cloud computing. Using Cloud environment all the applications can be delivered as a web service. Cloud enables the delivery of applications, software development platform and servers/ hardware as a service. Size of the data to be processed by these applications is increasing exponentially. The large applications are processing the data of several petabytes to exabytes range. Processing such a huge datasets is beyond the control of traditional database and analytics applications. This size of data imposes several challenges to the traditional environment including operating systems/ middle-wares, file systems, databases and data processing applications. These challenges include storing the data, searching the data, transfer and analysis of the datasets. To process these datasets newer or enhanced operating system/ middle-ware, file systems, databases and analysis applications have been evolved. Big data refers to a data set whose processing is beyond the capacity of traditional datasets. Big data requires different hosting platforms, different storage file systems, different programming models and different databases. The hosting environment is generally the distributed servers or cloud based hosting platform [20] [14]. The special newer file systems have been designed to store very large files like Google file system or Hadoop distributed file system [18], [21]. New programming model like Map-Reduce have been evolved to process these datasets. Finally the new schema-free databases have been developed to provide database support to these applications like MongoDB [15].

Generally, these datasets are used for analysis or semantic processing. The applications of big data analytics ranges from a large number of verticals including scientific and commercial applications including security analysis (intrusion detection, high performance cryptography, threat detection etc.), searching and mining (large scale social media system, recommendation systems, link and graph mining etc.), healthcare analysis, financial analysis, telecommunications analysis etc.

The rest of this paper is organized as follows. In Section 2, we present the problem statement of the paper. In Section 3, we described the proposed Cloud and Big Data based Higher Education Quality Enhancer Framework (CHEQEF). Section 4, presents the experimentation and discussion on result analysis. Related work is discussed in Section 5. Finally, Section 6 concludes the paper and proposes future work.

2 Problem Statement

The focus of this paper is to apply the cloud based big data analytics to enhance the quality of higher education. We have collected the data related to student's results and learning from universities of Saudi Arab and India for couple of years. We are considering this problem as big data problem as the couple of years result is also a huge data for all the students of hundreds of higher education institutions. Generally, the there are several criteria that can be used for setting quality parameters - statistics, indicators and benchmarks. These criteria are grouped into seven categories as: (1) Curricular aspects, (2) Teaching, learning and evaluation, (3) Research, consultancy and extension, (4) Infrastructure and learning resources, (5) Student support and progression, (6) Governance and leadership, and (7) Innovative practices. Based on the data available from the other universities we can

utilize the big data analytics to find the level of a university in comparison with the others and the area where the university needs to improve upon. In this research we are mainly focusing on category 2, category 4 and category 5. We have collected the student results of a number of higher educational institutions.

To improve the quality of higher education one should focus on the better learning experience. If the learning is proper, it will have positive impact on entire teaching learning experience. To improve overall learning experience the higher education institutions should focus on in-depth analysis of results to find the area of weakness of students. After identification of the subjects the emphasis should be given to determine the reason behind the weakness. There are numerous research works on application of big data in higher education system. We feel that these research works are in preliminary stage. However, to the best of our knowledge, based on the exhaustive literature survey there is no efforts made to use big data to identify the weak subjects and seek the way to improve it. Moreover, none of the research has focused on improving the quality of higher education by adopting the cloud and big data based recommendation system. To fill this gap we are proposing and developing Cloud and Big Data based Higher Education Quality Enhancer Framework (CHEQEF) discussed in next section 3.
We propose to use open source tools as numerous open source tools are available that supports all types of processing of big data. These tools include Cloud computing tools and big data analytics. These tools belongs to various category including IaaS middleware, large scale file systems, large scale database, programming environments, analysis or reporting tools etc. We are using Opennebula and Openstack as IaaS middleware, HDFS as files system, Hadoop as a MapReduce programming environment and JasperReport/ R as reporting tool [11], [12], [17], [16], [19], [13].

3 Cloud and Big Data based Higher Education Quality Enhancer Framework (CHEQEF)

This framework is using the examination results and feedback of various stack holders for proposing the changes in subject scheme in a semester or year and the syllabus of some subjects. These proposed changes in subjects and syllabus will ultimately results in overall learning experience of students and leads to a successful future. The framework performs the recommendation computation in multiple steps. It initiates the processing by analyzing the vast data of students results of a number of years. This step generates year wise list of subjects with average marks of all the students who has taken examination for this subject. This result will further analyzed to prepare multiple list of subjects having good marks in each year, similar year wise list of subject having average marks and below average marks. This second list will help in identifying the subjects having good scoring, average and low scoring trend. With the help of this list we can identify which subjects are generally low or average scoring subject and require special attention. This step will also generate another list with the details of students with good or low scoring in each subject. This list will help in identifying subject wise low scoring students. This list will be one input for next step.

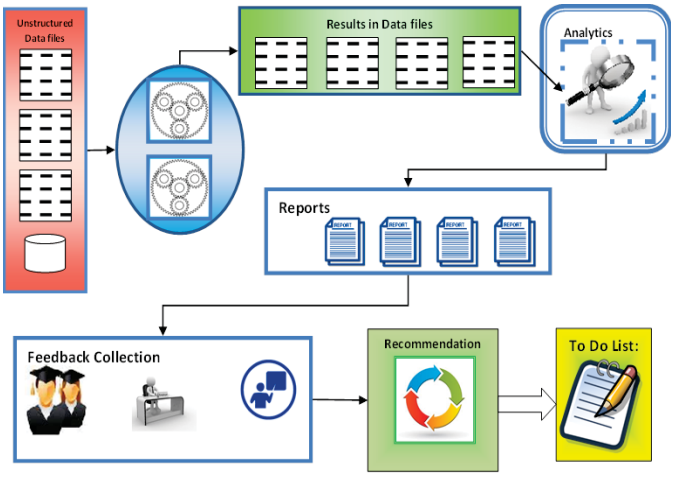


Figure 1. System flow of CHEQEF

In the next step we take feedback of low scoring students in the specific subjects obtained in the previous steps. This feedback management system is integrated with the framework to automate the feedback collection process. The system mainly seeks the answer to why their score is low is it? Not of their interest, beyond their capability, syllabus is lengthy, any specific part is problematic, the subject requires more gaps during the exams etc. This student feedback and the low/ average scoring subject list produced in previous step will be passed to the next step as input for further processing.

The next step is to collect the feedback of another stockholder that is the delegates from industry employing these students. We seek the industries recommendation on various subjects and their syllabus. The industry delegates will be presented will all the statistics generated so far. The purpose of taking inputs from industry is to analyze that whether the current subject scheme and combination is as per the industry demand. There might be a possibility that students are scoring well in some subjects but these subjects are not the demand of industry. If this is the case, then reducing syllabus of such subjects/ changing the subject, may help students to find more time for better preparation of low scoring subjects.

Now, the framework seeks the feedback of last and most important stakeholder of teaching learning process that is faculty members. The result analysis data, student feedback and industry feedback is given to the faculty members and their feedback on the overall situation is recorded. Finally, the framework recommends the changes in subject scheme and syllabus of some subjects.

The system flow diagram of the framework is depicted in Figure 1. The process starts with the analysis of results of several years. The purpose of result analysis is to find the scoring trend of different type of subjects (for example theoretical/ practical oriented-for science, similarly theoretical subjects/ numerical subject in commerce/ management, on the similar track classroom training/ field work based subject). The scoring may be good or high or average to low. This analysis will results in a list of satisfactory and dissatisfactory subjects. Now, we have two options, the first option is go for further analysis that is explained in next paragraph and covered in this paper. The second option is, if for a subject overall experience of a number of years that it is low scoring. In such situation we can stop further processing and we can try a simple solution for example increase the gap between the examinations of such papers, if this doesn't work then further fine grain the result analysis for each question from examination answer copy to focus on the specific weak area from syllabus and that part of the syllabus can be removed or more emphasized. At present we are collecting data at this level for further analysis in future.

Now the system needs further input from industry where the students are being adopted with list of satisfactory and dissatisfactory subjects. The industry feedback should reflect whether the present situation is fine or they expect students to perform better in dissatisfactory subjects and if some of satisfactory subjects are not desired according to the current need of the industry. Moreover, this framework also expects inputs from students that seek answer to why they have dissatisfactory scoring in some subjects? Is it the limitation of their capability or the subject is not of their interest (for example the students are interested in numerical/practical/field work/classroom oriented or theoretical subjects). At last the framework also seeks the feedback of the other core part of learning experience that is the faculty members. Faculty members presented with the overall situation and the feedback of the other stakeholders.

4 Experiments and Analysis of Results

The data obtained from the various source is very diverse in nature as from some institutions we got the data as un structured data files for some other system we got the data in relational database. These database are also heterogeneous as they are using different vendor database software and with different schema. Mostly it is in the form of unstructured files hence, we have converted the data in homogeneous form with uniform schema in test files form. We performed the analysis experiments in a lab having Hadoop cluster of 15 high end computers. These computers are equipped with 32 GB RAM, 1 TB hard-disk and i7 processor. Besides the Hadoop based map reduce program in Java. We also analyzed the data using Hive queries for cross checking of our results.

Graph in Figure 2 is the result of processing performed on the final result of first year science students for the 2013 and 2014 examination of bachelor degree. The result data is obtained from 150 higher education institutions. From the graph, it can be observed that the student scoring for two years is almost similar in terms of the scoring level. For example, the average percentage marks of one subject are higher for both the year and if lower, then for both the year the score is low. We also analyzed student result for 2012 from the same institutions and the trend in scoring level is also similar for the third year. Initially we only used 4 systems in the cluster and verified the results of our application and hive queries. As the program is doing well, we started collecting data for around 10 years and from more universities and institutions.

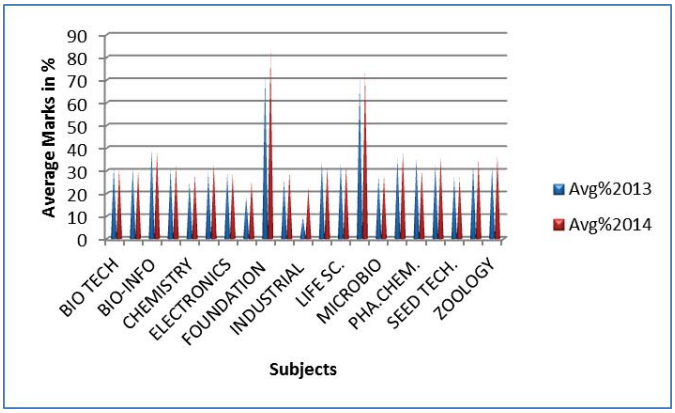


Figure 2. Year wise average percentage of different subjects

Now, this result is presented to the stakeholders for their further feedback. After getting feedback from industry, students and faculty the system presents the report in the form of suggestion for reducing syllabus, changing the scheme and increase the gap between the two papers in the examination. At present we are collecting feedbacks on the result analysis.

5 Related Work

Literature in the area of applying cloud based big data in higher education contains a few and very recent research works. The focus of this literature survey is the use of big data analytics in higher education.

Liang-qiu and Liang-qiu [1] have discussed the definition and features of big data and its applications in higher education.

As per the finding of the paper, the applications of big data in higher education are in its very preliminary stage. They also discussed a number of fundamental problems to be solved and questions to be answered before adopting big data in higher education. Their work mainly presents four applications as forecasting, providing a new platform for education, development of new teaching mode and promoting the exchange of research and innovation.

West [2] has given detailed insights on using big data for education with the help of data mining, data analytics and web dashboards. The author has examined the potential for improved research, evaluation and their accountability through the use of big data. The author also discussed how to tackle the operational and policy based obstacle in adopting big data for education. The report also discusses a number of web dashboards.

Ben and Russell [3] have presented their in-depth understanding of big data literature and method to utilize the vast amount of available data from higher education institutions. This research starts with the basics of big data, then moves towards the applying big data in higher education. The authors also cover sources and types of big data in higher education. They also presented various opportunities and challenges of implementing big data analytics in higher education. The research work concluded by presenting a number of future directions. The paper suggests involvement of interaction among various higher education stack holders in improving the overall experience of higher education.

Leah et al. [4] presents their view on how to realize the excellent offering of big data and learning analytics for higher education. The papers discusses that to effectively realize it's the changes in culture, infrastructure and various teaching practices including assessment, learning cannot be achieved just by implementing the new big data tools. The authors have also presented various associated challenges for the desired change in the educational system.

Tristan [5] has presented various challenges faced by students of initially degree program mostly caused by the change in the curricular structure. The author has discussed the role of predictive analytics and choice architecture to effectively manage this situation. Author has also examines a course recommendation system named as Degree Compass. Degree compass analyses student's capability and available courses. This system pairs the student's capability with the courses available for upcoming sessions.

According to Maarten and Fleur [6], recent higher education challenges and trends demand for reorientation towards student interaction. They have introduced social learning analytics (SLA) as a tool for informative assessment practices. The tool is focused at promoting student as active learning in open and social learning environment. The paper uses the data available for student's social connectivity and activity for assessment practices.

Christina et al. [7] have presented results of their study on how part time employment of students in student affairs division positively affects student development in preparation for their future. As per their finding this employment offers and environment to student where they can apply their knowledge gained so far and enables them to acquire new competencies. The authors used online survey as a tool to perform this study. They have concluded that impact of other jobs inside and outside of university to the students' progress.

Ming and Mengxing [8] have discussed need of innovation in teaching methods and kind of thinking ability with the focus of future societal demand. The authors explored the possibility of reforms in pedagogy in higher education with the help of big data concepts and technologies. The authors have also discussed a number of flaws in Chinese college teaching and proposed ways to improve teaching methods accordingly.

Jeffrey [9] has presented the ethics of using big data for higher education. The author starts with highlighting the benefits offered by the big data and data mining techniques. The paper also highly recommends a number of ethical challenges to be taken care of while applying the big data and data mining to educational system.

Candace et al. [10] discusses the differences in the assessment process while the assessments with the large scale data from the online learning environment and assessment without such data. The paper explains the discussion with the help of three case studies from different kind of learning platforms including first environment with interactive exercises, second online environment having programming practice, and a massive open online course (MOOC).

6 Conclusion and Future Work

This paper presents a cloud and big data based framework for overall quality improvement in higher education. The framework helps in improving higher education quality recommending changes in curriculum and scheme. The main focus of the framework to identify weaker area of subjects, reason behind the weakness and improving the syllabus and overall scheme by considering inputs from all stakeholders of higher education system including students, faculty members and employers. After detailed analysis this framework works as recommender system and suggests the changes in the course scheme and syllabus. We feel that adopting these changes will results in better learning experience and ultimately positively impact the education quality.

Along with the exhaustive analysis of results, online feedback by students, faculty, employer. We have also started working on incorporating the social media activities of the stakeholders. As suggested in the last of fourth paragraph of section 3, we will also further analyze the results for at least a decade at the question level finer grain. We have started data collection in this reference.

Acknowledgements

This work has been supported by King Faisal University, Ministry of Higher Education, Kingdom of Saudi Arabia.

7 References

[1] Liang-qiu Meng, & Liang-qiu Meng "Application of Big Data in Higher Education" In 2nd International Conference on Teaching and Computational Science. Atlantis Press. (May, 2014).

[2] Darrell M. West "Big data for education: Data mining, data analytics, and web dashboards. Governance Studies at Brookings", http://www.brookings.edu/~/media/Research/Files/Papers/2012/9/04 education technology west/04 education technology west.pdf(Sept 2012).

[3] Ben Daniel, & Rusell Butson "Foundations of Big Data and Analytics in Higher Education. In International Conference on Analytics Driven Solutions" ICAS2014 (p. 39). Academic Conferences Limited. (Sept 2014).

[4] Leah P. Macfadyen, Shane Dawson, Abelardo Pardo, & Dragan Gasevic "Embracing big data in complex educational systems: The learning analytics imperative and the policy challenge. Research & Practice in Assessment", Vol. No 9., Issue No 2, Page number (17-28), (2014).

[5] Tristan Denley "How predictive analytics and choice architecture can improve student success. Research & Practice in Assessment", Vol. No 9., Issue No 2, Page number (61-69)(2014).

[6] Maarten de Laat & Fleur R. Prinsen, "Social Learning Analytics: Navigating the Changing Settings of Higher Education" (2014).

[7] Christina Athas, Lance Kennedy-Phillips, & D'Arcy John Oaks, "Student employee development in student affairs. Research & Practice in Assessment".

[8] Ming, K., & Mengxing, L. "Innovative Thinking in Collegiate Pedagogy in the Big Data Era—Analysis of the Teaching Platform Required in China. Chinese Studies", Col No. 3., Issue No, 4, Page numbers (136-138) (2014).

[9] Jeffrey Alan Johnson, "The Ethics of Big Data in Higher Education. International Review of Information Ethics",Vol 7,Page numbers(3-10) (2014).

[10] Candace Thille, Emily Schneider, René F. Kizilcec, Christopher Piech, Sherif A. Halawa, and Daniel K. Greene. "The Future of Data–Enriched Assessment. Research & Practice in Assessment" Vol No. 9, Issue N0. 2 page numbers (5-16) (2014).

[11] [OPENNEBULA] Opennebula IaaS Toolkit, Available Online at: http://opennebula.org/.

[12] [OPENSTACK] Openstack IaaS Toolkit, Available Online at: http://www.openstack.org/.

[13] [R] The R Programming Language, Available Online at: http://www.r-project.org/.

[14] [Rackspace] Rackspace Infrastructure as a Service, Available Online at: http://www.rackspace.com/.

[15] [MONGODB] Mongo Database, Available Online at: http://www.mongodb.org/.

[16] [MAPREDUCE] Map Reduce Programming, Available Online at: http://en.wikipedia.org/wiki/MapReduce.

[17] [HADOOP] Hadoop Toolkit, Available Online at: http://hadoop.apache.org/.

[18] [GFS] Google File System, Available online at: http://en.wikipedia.org/wiki/Google_File_System.

[19] [JASPER] Jasper Reports and Business Intelligence, Available Online at: https://www.jaspersoft.com/reporting.

[20] [AMAZON] Amazon Cloud Services, Available Online at: http://www.amazon.com.

[21] [HDFS] Hadoop Distributed File System, Available online at: http://hadoop.apache.org/docs/stable/hdfs_design.html.

SESSION

BIG DATA SEARCH, MINING METHODS, CLUSTERING, KNOWLEDGE EXTRACTION AND ENGINEERING

Chair(s)

TBA

Randomized Sorting as a Big Data Search Algorithm

I. Avramovic and D. S. Richards
Department of Computer Science, George Mason University, Fairfax, Virginia, USA

Abstract—*A search problem may be formulated in which a goal state is sought, using comparisons of data elements against a large database to evaluate progress towards the goal. Given a big data aided algorithm for such a problem, we ask whether the algorithm converges to the desired goal, and if not, what can be said about the state at which the algorithm terminates. converges. To this end, we present a randomized sorting algorithm called consensus sorting, which makes use of queries to a large database of permutations. The algorithm refers decisions on randomly selected pairs of elements to the database, which uses a consensus of permutation scores to determine whether swapping is a good sorting decision. Depending on the scoring method used, the algorithm converges with differing degrees of sortedness. The convergence behavior is described, and an explanation of the behavior is formulated.*

Keywords: sorting, randomized algorithms, searching, combinatorics, simulation and modeling

1. Introduction

Consider a search problem where we are given a starting state, and seek a goal state. Progress towards the goal is made through a series of movements, and the decision for each movement is evaluated by an appeal to a large database. In other words, our sense of position is derived from the database. We would like to know how close we can come to the goal via this process. This can alternatively be phrased as a negative, asking whether successive appeals to a large database will bias a result towards one particular value.

As an example, consider the case where one has a photo, and wants to determine the location where the photo was taken. The state is a set of potential locations of the photo, which has a starting state that includes the entire world, and a goal state which is the location that the photo was taken. Each movement towards the goal is a refinement of the set of locations, by comparing the image for similarity, against a large database of images from around the world. This general approach was implemented in the Im2gps algorithm presented by Hays and Efros [4].

Increased use of information sharing through Cloud computing has led to increased volumes of available data [3], thus big data searches which follow the same formulation may become increasingly common. Data available on the Cloud tends to have decreasing informational value density but increasing overall value [6], and may have restrictions on the nature of access to the available data. Thus, a user who wants to leverage the value of the available Cloud data may find themselves using a search mechanism similar to the one described here.

This paper will consider a more abstract version of the problem. This is done for the sake of analyzing the behavior as the state approaches the goal state, and better answering the question of how closely the goal can be determined through this approach. The formulation considered here is a randomized sorting algorithm, where the state is a permutation, and the goal state is the identity permutation, representing a sorted array. Each movement step involves selecting a random pair of elements from the permutation, and comparing against the database to determine whether swapping the pair of elements is a good sorting decision. The algorithm shall be referred to as a consensus sorting algorithm.

Ideally, the consensus sorting algorithm will result in a fully sorted array, but due to the contents of the database and the nature of the swap criterion, there are conditions where this does not occur. The swap decision made by the algorithm is based on a large database of sample permutations, as well as a scoring function which heuristically measures the "sortedness" of a permutation. By comparing against the scores of permutations in the database, the algorithm finds a consensus on whether it would be better to swap the elements or to preserve their order. The process is then repeated, with the hope that the permutation will eventually reach a fully sorted state. Empirically, the larger the database that is used, the better the sorting result, although the most significant factor is the scoring function used by the database.

From a practical perspective, consensus sorting does not have a good run-time, and random error introduces the likelihood of imperfect sorting results. What is of interest is the unexpected observation that the sorting process of this algorithm may converge to a point which is different than a fully sorted array. This paper will analyze this convergence phenomenon, and explores its cause.

Analysis centers around the scoring function which is used, and the way in which it impacts the algorithm's capability to sort. Indeed, some choices of scoring functions do not lead to a sorted array in experimental trials. For example, a scoring function based on the number of cycles in the permutation will return a nearly sorted array in most cases, while scoring by number of inversions in the permutation will sort a fraction of the way, and then stop progressing.

This paper will describe the consensus sorting algorithm in detail, and describe examples of scoring functions which can be used with the algorithm. It then discusses the circumstances under which the sorting algorithm stabilizes, and finally analyzes aspects of the inversion metric, one of the metrics used for scoring.

2. Consensus Sorting Algorithm

We begin by specifying the consensus sorting algorithm, for which some definitions will be needed. Assume that the input array is a permutation $\pi \in S_n$ of n elements. The terms array and permutation may be used synonymously in this context. We are given a *permutation database* $\mathcal{D}$ which is a collection of permutations (only the permutations of length n in $\mathcal{D}$ are relevant), along with a scoring function $\phi : S_n \to \mathbb{N}_0$. The scoring function must satisfy the condition that a score of zero uniquely corresponds to an identity permutation (a fully sorted array). Scores should decrease to reflect greater "sortedness".

Definition An *ideal database* for a permutation length n is a database $\mathcal{D}_I$ in which every permutation of length n is represented an equal, nonzero number of times.

Every database considered in this paper shall either be ideal, or shall be chosen by independent random selection with replacement from the set of all permutations of the appropriate length.

Definition Define a *consensus swapping function* as follows:
$F_{\phi,\mathcal{D}} : \mathbb{Z} \times \mathbb{Z} \times S_n \to S_n$, which takes two integers i and j, with $1 \leq i, j \leq n$, as well as a permutation π_a. It returns the permutation π_{a+1}, which represents the next step in sorting. If an ideal database is used, the function may simply be denoted F_ϕ.

For a given π_a, let $\mathcal{C}_M$ be the collection of all permutations of length n from $\mathcal{D}$ in which the elements at positions i and j match the elements in positions i and j of π_a, respectively. Likewise, let $\mathcal{C}_R$ be the collection of all permutations of length n from $\mathcal{D}$ in which the elements in positions i and j match the elements in positions j and i of π_a, respectively (the reverse order).

Let μ_M and μ_R be the arithmetic mean of the scores of every member of $\mathcal{C}_M$ or $\mathcal{C}_R$, respectively. The consensus swapping function can be specified as follows.

$$\mu_M = \frac{1}{|\mathcal{C}_M|} \cdot \sum_{\pi \in \mathcal{C}_M} \phi(\pi) \tag{1}$$

$$\mu_R = \frac{1}{|\mathcal{C}_R|} \cdot \sum_{\pi \in \mathcal{C}_R} \phi(\pi) \tag{2}$$

$$F_{\phi,\mathcal{D}}(i,j,\pi_a) = \begin{cases} \pi_a & \text{if } \mu_M \text{ or } \mu_R \text{ is undefined,} \\ & \text{or } \mu_M \leq \mu_R, \\ & \text{or } i = j, \\ (i\ j) \cdot \pi_a & \text{otherwise.} \end{cases} \tag{3}$$

Here, $(i\ j) \cdot \pi_a$ denotes π_a with the elements in positions i and j swapped.

The algorithm proceeds as follows, given an array π_a. A pair of distinct positions i, j are selected from the array at random, with $i \neq j$.[1] The next iteration of the array is computed as $\pi_{a+1} = F_{\phi,\mathcal{D}}(i,j,\pi_a)$. The process is repeated until some stabilization criterion is satisfied.

The criterion used in this paper to determine stabilization is the lack of improvement in score differential over a fixed interval, in other words, when $\phi(\pi_a) - \phi(\pi_{a-k}) \geq 0$ for some fixed integer $k > 0$. This criterion is guaranteed to terminate, because from any starting point, only a finite number of score improvements are possible without at least one interval which is not an improvement.

The algorithm is displayed as pseudo-code below.

while stabilization condition unsatisfied for π_a **do**
  $i \leftarrow random(1, n)$
  $j \leftarrow random(1, n-1)$
  if $i = j$ **then**
    $j \leftarrow n$
  end if
  $\pi_{a+1} \leftarrow F_{\phi,\mathcal{D}}(i,j,\pi_a)$
  $a \leftarrow a + 1$
end while

Consider a pair of distinct elements $\pi_a(i)$ and $\pi_a(j)$ from some permutation π_a, and assume WLOG that $i < j$. When discussing the ordering of the pair of elements, several different terms may be used depending on which notion of ordering is under consideration.

Definition If $i < j$ and $\pi_a(i) > \pi_a(j)$, the pair of elements are said to be *inverted*. If $i < j$ and $\pi_a(i) < \pi_a(j)$, they are *in order*.

Definition Given a second permutation π', if $\pi'(i) = \pi_a(i)$ and $\pi'(j) = \pi_a(j)$, then π' is said to be a *matching* permutation relative to π_a given i and j. If $\pi'(i) = \pi_a(j)$ and $\pi'(j) = \pi_a(i)$, then π' is said to be a *reversed* permutation.

Definition Let $\pi_{a+1} = F_{\phi,\mathcal{D}}(i,j,\pi_a)$ be the sorting step following π_a, given scoring metric ϕ and database $\mathcal{D}$. If $\pi_{a+1} = F_\phi(i,j,\pi_a)$, then π_{a+1} represents a *correct* sorting step following π_a, otherwise it represents an *incorrect* sorting step.

[1]The algorithm may be relaxed to allow $i = j$, in which case the array order would remain unchanged in the instances when an equal pair was selected.

Informally, inverted and in order elements refer to ordering with respect to a sorted array, matching and reversed permutations refer to ordering with respect to a reference permutation, and correct and incorrect permutations refer to ordering with respect to a scoring metric.

3. Scoring Functions

There are a number of possible scoring metrics that are common in adaptive sorting literature[2], and can be used with the consensus sorting algorithm. Not all metrics are equally effective for sorting purposes, and a number of metrics which exist are unlikely to produce a sorted array. Two which are successful in sorting, to a degree, are inversion counting, and cycle counting. Using a randomly selected database and the inversions metric, ϕ_{inv}, the score will stabilize at a fraction of the maximum value, depending on database size and array size. The cycle count metric, ϕ_{cyc}, regularly reaches a sorted or nearly sorted state. Below are some of the metrics which were tested, using a permutation length of $n = 100$ and a database of $|\mathcal{D}| = 1,000,000$ permutations, with an epoch of 50 runs.

1) The *number of inversions*, ϕ_{inv}, in the array. An inversion is any pair of elements in the array which are out of order. Using a completely arbitrary permutation, the inversion metric tends to initially improve the score through sorting, until the score reaches a point which is roughly half the original. Once it reaches that point, the score will oscillate around the point without significantly improving further. Beginning instead with an identity permutation (a completely sorted array), the score will begin to rise under this metric, until it stabilizes near the same value referred to above.
2) The *reverse cycle count*, ϕ_{cyc}, of the array, which is defined as $n - c$, with c being the total number of cycles in the array. Sorting by this metric regularly reaches a completely sorted or nearly sorted (score of 1 or 2) state.
3) The *size of the largest inversion*, ϕ_{li}, in the array. This is defined as the maximum of the differences between the indexes of pairs of inverted elements in the array, or 0 if there are none. Beginning with an arbitrary permutation, the score does not reduce significantly from its original value by using this metric.
4) The *largest distance between an element and its correct slot*, ϕ_{ld}. As with the largest inversion, the score does not reduce significantly by using this metric.
5) The *minimum number of removals for a sorted subsequence*, ϕ_{ss}. This represents the minimum number of elements that would need to be removed from the array to leave behind a sorted subsequence of the original array. Although this is intuitively a stronger measure of sortedness than the largest distance metrics, is also fails to reduce score effectively.
6) The *number of runs*, ϕ_{run}, in the array. This is determined by the number of boundaries between runs, which is defined as the number of elements that are out of order relative to the preceding element in the array. The score does not reduce significantly with this metric.
7) The *minimum number of ascending subsequences*, ϕ_{asc} in the array. This is the minimum number of subsequences, minus one, into which the elements of the array can be portioned such that each subsequence is ascending. The score does not reduce significantly with this metric.

Table 1: A comparison of the change in score for consensus sorting over different scoring metrics.

	metric	max	start score	final score
ϕ_{inv}	number of inversions	4950	50.0%	25.9%
ϕ_{cyc}	cycle count	99	95.8%	1.15%
ϕ_{li}	largest inversion	99	99.2%	95.8%
ϕ_{ld}	largest distance	99	92.3%	85.6%
ϕ_{ss}	sorted subsequence	99	92.4%	91.1%
ϕ_{run}	number of runs	99	50.0%	51.0%
ϕ_{asc}	ascending subsequences	99	15.9%	13.9%

The results of the tests are shown in table 1. The table gives the maximum possible score for each of the metrics shown, along with the mean score of an arbitrarily chosen permutation and the mean final score of the permutations. The starting score and final score are given in terms of a percentage of the maximum score, in order to normalize for differences in metric. The metrics ϕ_{inv} and ϕ_{cyc} were the most effective in sorting, while the remaining metrics showed negligible improvement. The low final scores for ϕ_{asc}, and to a lesser degree ϕ_{run}, are misleading, because they have a relatively low starting score as well.

We shall focus on the inversion metric, ϕ_{inv}, because it is straightforward, and because its partial effectiveness in sorting makes it interesting. We begin by making an observation about the sorting capability of a scoring metric.

Given some scoring metric ϕ and database $\mathcal{D}$, suppose that $\phi\left(F_{\phi,\mathcal{D}}(i,j,\pi)\right) \leq \phi(\pi)$ for all legal values of i, j, and π - in other words, the score is monotonically non-increasing as a result of iterating the consensus sorting algorithm. If every non-sorted array has some legal choice of i and j that reduces the score when swapped, then the algorithm will give a fully sorted array if the number of iterations is allowed to increase without bound. This is because the permutation score has a finite number of legal values. The values do not increase, and have a lower bound of zero. If for each iteration, the minimum chance that the score decreases is fixed and non-zero for an unsorted array, then eventually the score will reach the value of zero.

The theoretical guarantee of a sorted array applies mostly to situations when an ideal database is used, because if a randomly selected database is used, it is unlikely that the

conditions of the guarantee will hold. However, if an ideal database is used, the scoring function of the inversion and cycle metrics can be shown to produce monotonically non-increasing results, and a score-reducing selection always exists for arrays which are not fully sorted. Thus, both metrics will theoretically produce sorted arrays.

4. Steady-State Hypothesis

In theory, using an ideal database with either the cycle metric or the inversion metric should result in a fully sorted array. Thus it may be perplexing that the experimental results show a different convergence behavior when an ideal database is not used. This is particularly true of the inversion metric, where sorting ceases to be effective well short of the expected array. Clearly this is due to the fact that the database was chosen randomly rather than using $\mathcal{D}_I$, but the way in which the different database impacts the result remains to be determined.

Definition An *error* is any incorrect swapping decision made due to the chosen database[2], that is, whenever the active database makes the opposite swapping decision compared to an ideal database for the same input permutation and same choice of indices.

This leads to the conclusion that at certain times, the impact of error on the score is sufficient to prevent the score from reducing as much as theoretically expected.

Definition In general, a reduction in score from one iteration to the next is referred to as an *improvement* in score, while an increase in score (or a negative improvement) is referred to as a *degradation* in score. Thus, the improvement in score between a permutation π_a and $\pi_{a+1} = F_{\phi,\mathcal{D}}(i,j,\pi_a)$ is given by $-[\phi(\pi_{a+1}) - \phi(\pi_a)]$, where i and j is the selected element pair.

Definition For a given ϕ and randomly chosen database $\mathcal{D}$, we describe the *mean improvement* $I_{\phi,\mathcal{D}}(s,n)$ of all permutations π of length n with a score of $s = \phi(\pi)$. For a given permutation π, let the mean score improvement of all subsequent iterations be denoted by:

$$\Delta\hat{s}(\pi) = \frac{1}{n(n-1)} \cdot \sum_{j=1}^{n}\sum_{i=1}^{n} -\left[\phi\left(F_{\phi,\mathcal{D}}(i,j,\pi)\right) - \phi(\pi)\right] \tag{4}$$

Then, letting $\mathcal{C}_{s,n} = \{\pi \in S_n \mid s = \phi(\pi)\}$,

$$I_{\phi,\mathcal{D}}(s,n) = \frac{1}{|\mathcal{C}_{s,n}|} \cdot \sum_{\pi \in \mathcal{C}_{s,n}} \Delta\hat{s}(\pi) \tag{5}$$

In the case of an ideal database, the simplified notation $I_\phi(s,n)$ will be used.

[2]Recall that correctness is defined relative to sorting decisions made using an ideal database.

If a random database is used, the mean improvement will differ from the mean improvement for an ideal database due to the presence of errors. We can express this principle by stating that, for some $N_{\phi,\mathcal{D}}(s,n)$, which shall be called the improvement *noise* due to error,

$$I_{\phi,\mathcal{D}}(s,n) = I_\phi(s,n) + N_{\phi,\mathcal{D}}(s,n) \tag{6}$$

Definition The *steady-state hypothesis* is the claim that for a given function ϕ and database $\mathcal{D}$, the consensus sorting algorithm will tend to stabilize near a score s, which is determined by the point at which $I_{\phi,\mathcal{D}}(s,n)$ is closest to zero. The score s around which the algorithm stabilizes is the *equilibrium* score.

Intuitively, for as long as the score is expected to improve, on average, it probably will, and as long as the score is expected to degrade, on average, it probably will. Thus, it is logical that an equilibrium point would occur where the score neither improves nor degrades on average.

It stands to reason that if $I_\phi(s,n)$ can be computed and $N_{\phi,\mathcal{D}}(s,n)$ estimated for a scoring function ϕ, they can be used in equation 6 to estimate equilibrium score, where the algorithm is expected to converge.

5. Inversion Metric

This section analyzes the inversion metric, so unless otherwise noted, assume that $\phi = \phi_{inv}$.

Given a permutation π of length n, a pair of indices i and j, and elements $\pi(i)$ and $\pi(j)$, let $m_{i,j}(\pi) = m_{j,i}(\pi)$ be the number of elements on the interval from i to j (assuming WLOG that $i < j$) whose value is greater than $min(\pi(i),\pi(j))$ and smaller than $max(\pi(i),\pi(j))$. Swapping the elements at i and j will result in a change in score of $2m_{i,j}(\pi)+1$, because the only elements which have an effect on the number of inversions are these mid-value points, as well as the pair being swapped itself. Whether the change is an improvement or degradation depends on whether i and j were inverted or in order to begin with.

The fact that the inversion metric produces a monotonically non-increasing scoring function for an ideal database is a consequence. For any pair of indices i and j and element values $\pi(i)$ and $\pi(j)$, there is a one-to-one correspondence between permutations with i and j in order and permutations with i and j inverted. Since each in order permutation has a lower score than its corresponding inverted permutation, the consensus will also be to put i and j in order.

Defining $R_m(\pi)$ as the number of selections of i and j which are inverted with $m_{i,j}(\pi) = m$, the expression in 4 can be regrouped by m value, and rewritten as the following.

$$\Delta\hat{s}(\pi) = \frac{1}{n(n-1)} \cdot \sum_{m=0}^{\infty} R_m(\pi) \cdot (2m+1) \tag{7}$$

For the given n, let $A_{n,m}(z)$ be a generating function which has exponents that correspond to permutation score,

and coefficients which correspond to the number of selections of i, j, and π such that $|\pi| = n$ and $m_{i,j}(\pi) = m$. Let $[z^s]A_{n,m}(z)$ be the coefficient of z^s in $A_{n,m}(z)$. Furthermore, let $A^{(ord)}_{n,m}(z)$ be a generating function which only counts selections of i and j if they are in order in π. Similarly, let $A^{(inv)}_{n,m}(z)$ be a generating function which only counts i and j if they are inverted in π. Thus,

$$A_{n,m}(z) = A^{(ord)}_{n,m}(z) + A^{(inv)}_{n,m}(z) \tag{8}$$

$$\sum_{\pi \in \mathcal{C}_{s,n}} R_m(\pi) = [z^s]A^{(inv)}_{n,m}(z) \tag{9}$$

$$|\mathcal{C}_{s,n}| = \sum_{m=0}^{\infty} \frac{[z^s]A_{n,m}(z)}{n(n-1)} \tag{10}$$

$$I_\phi(s,n) = \frac{\sum\limits_{m=0}^{\infty} [z^s]A^{(inv)}_{n,m}(z) \cdot (2m+1)}{\sum\limits_{m=0}^{\infty} [z^s]A_{n,m}(z)} \tag{11}$$

No closed-form solution was found for $A^{(ord)}_{n,m}(z)$, $A^{(inv)}_{n,m}(z)$, or $A_{n,m}(z)$, but an algorithm for computing them was used.

Let $\epsilon_{\mathcal{D}}(i,j,\pi)$ be the chance of an incorrect sorting step for i and j from some permutation π. The improvement noise for π shall be modeled as the chance of error due to the database contents, $\epsilon_{\mathcal{D}}(i,j,\pi)$, times the effect on the score due to the error, averaged over all choices of i and j. Whether an error results in a swap or not, the effect on score due to error will be a degradation of $2m_{i,j}(\pi)+1$, because even a lack of swap due to error would negate the improvement from the swap. Due to the symmetry of the choices of i and j, the expression for improvement noise for π is given as follows.

$$\begin{aligned}\Delta\hat{s}_N(\pi) &= \sum_{j=1}^{n}\sum_{i=1}^{n} \frac{-\big(2m_{i,j}(\pi)+1\big)\cdot\epsilon_{\mathcal{D}}(i,j,\pi)}{n(n-1)} \\ &= 2\sum_{j=i+1}^{n}\sum_{i=1}^{n} \frac{-\big(2m_{i,j}(\pi)+1\big)\cdot\epsilon_{\mathcal{D}}(i,j,\pi)}{n(n-1)}\end{aligned} \tag{12}$$

The improvement noise $N_{\phi,\mathcal{D}}(s,n)$ is then simply the improvement noise for π averaged over all suitable values of π.

$$N_{\phi,\mathcal{D}}(s,n) = \frac{1}{|\mathcal{C}_{s,n}|}\sum_{\pi\in\mathcal{C}_{s,n}} \Delta\hat{s}_N(\pi) \tag{13}$$

To estimate the chance of error $\epsilon_{\mathcal{D}}(i,j,\pi_a)$ for a given permutation π_a and pair i, j, consider the next correct sorting step π_{a+1}. Thus, i and j must be in order in π_{a+1}. Next, consider the collection of permutations in $\mathcal{D}$ which match π_{a+1}, and the collection of permutations which are reversed relative to π_{a+1} and thus have i and j inverted. An error occurs when the mean score of all permutations in the matching collection is greater than the mean score of all permutations in the reversed collection. Thus, if X_M is a random variable representing the mean score of the matching collection, and X_R is a random variable representing the mean score of the reversed collection, then $\epsilon_{\mathcal{D}}(i,j,\pi_a) = Pr(X_M > X_R)$. Suppose that X_M and X_R have means μ_M and μ_R, and variances $\frac{\sigma_M^2}{N_M}$ and $\frac{\sigma_R^2}{N_R}$, respectively. The Central Limit Theorem[5] claims that if a sufficiently large number N of independent samples are taken from a given distribution with mean μ and variance σ^2, then the mean of those samples will approach a normally distributed value with mean μ and variance $\frac{\sigma^2}{N}$. Thus,

$$X_M - X_R \sim \mathcal{N}\left(\mu_M - \mu_R, \frac{\sigma_M^2}{N_M} + \frac{\sigma_R^2}{N_R}\right) \tag{14}$$

giving the following

$$\epsilon_{\mathcal{D}}(i,j,\pi_a) = 1 - \Phi\left(\frac{\mu_R - \mu_M}{\sqrt{\sigma_M^2/N_M + \sigma_R^2/N_R}}\right) \tag{15}$$

where Φ is the cumulative distribution function of the normal distribution $\mathcal{N}(0,1)$. For the purposes of computation, the following approximation was used[1].

$$\Phi(x) \approx \begin{cases} \exp\left(-\frac{x^2}{2}\right)/12 + \exp\left(-\frac{2x^2}{3}\right)/4 & \text{if } x < 0, \\ 1 - \exp\left(-\frac{x^2}{2}\right)/12 - \exp\left(-\frac{2x^2}{3}\right)/4 & \text{if } x \geq 0. \end{cases} \tag{16}$$

Figures 1 through 6 in section 6 show a comparison between the point at which the sorting algorithm stabilized, and the theoretically predicted equilibrium point. The consensus sorting algorithm was run using the inversion metric, and the equilibrium point was averaged over an epoch of 500 runs, for array sizes ranging from 10 to 25, and database sizes ranging from 100 to $50,000$. The results are compared against theoretical predictions using the same set of array and database sizes.

The estimate does not represent a weighted average over all possible values of N_M and N_R, however reasonable results were obtained even using a single-value estimate. As can be seen in the results, the assumptions of the Central Limit Theorem used in the estimate no longer hold when the database size is small.

6. Equilibrium Point Results for Inversion Metric

Figures 1 and 2 show the results of running the consensus sorting algorithm and theoretically predicting the equilibrium point, respectively. The percentage on the vertical axis is the equilibrium score, represented as a percentage of the maximum possible score for a permutation of the given size. The tests were run for permutation sizes ranging from 10 to 25, and database sizes ranging from 100 to $50,000$. The equilibrium point for the algorithm was averaged over an epoch of 500 runs.

The results show an equilibrium point that decreases as database size increases and permutation size decreases. The empirical and theoretical results are similar, except for very small database sizes, where the prediction starts to break down. This is likely due to the use of the Central Limit Theorem to calculate the means from the data set. The Central Limit Theorem gives an approximation of the mean of samples from an arbitrary distribution, which becomes increasingly accurate for larger sample sizes (in this case, subsets of the database).

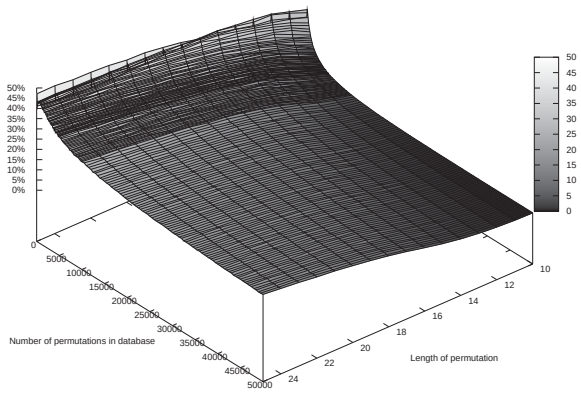


Fig. 1: The mean equilibrium point of the sorting algorithm, using a range of array sizes and database sizes.

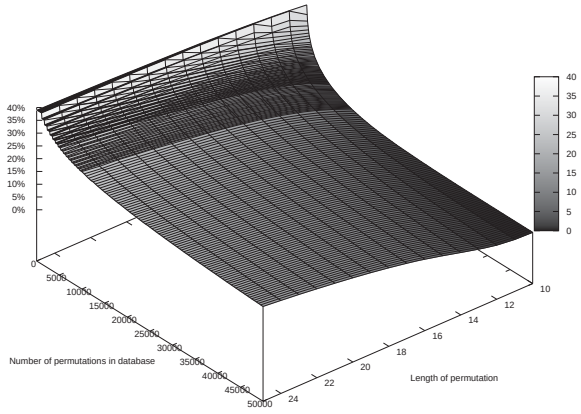


Fig. 2: The predicted equilibrium point of the sorting algorithm, using a range of array sizes and database sizes.

Figures 3 through 6 show a comparison of the empirical and theoretical results shown in the earlier figures, each for a specific value of the permutation size.

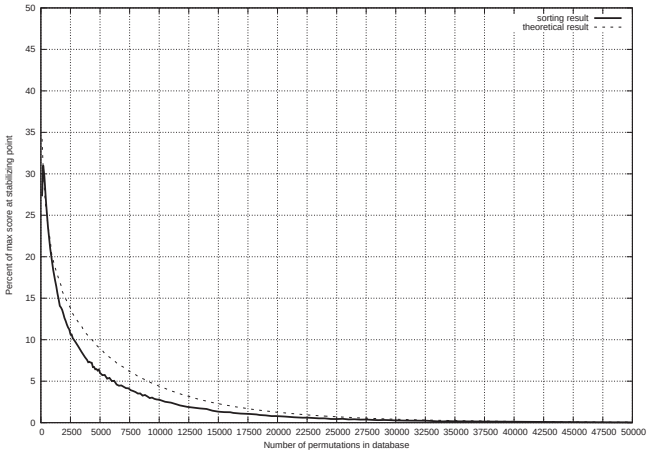


Fig. 3: Algorithm vs predicted equilibrium point, $n = 10$

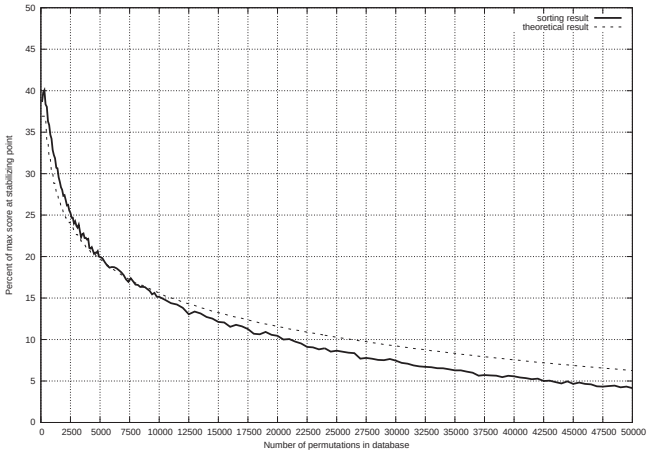


Fig. 4: Algorithm vs predicted equilibrium point, $n = 15$

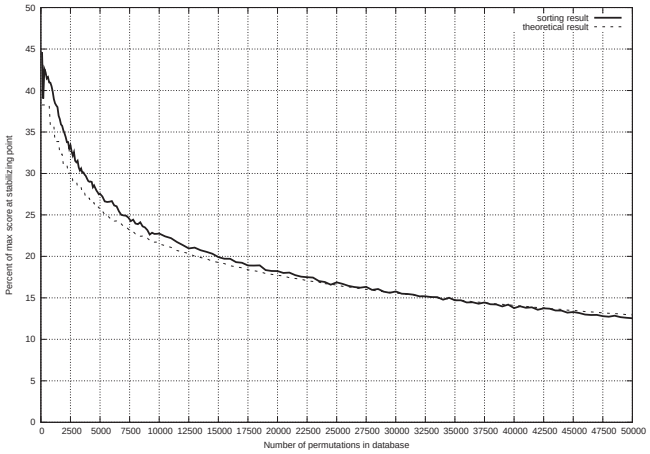


Fig. 5: Algorithm vs predicted equilibrium point, $n = 20$

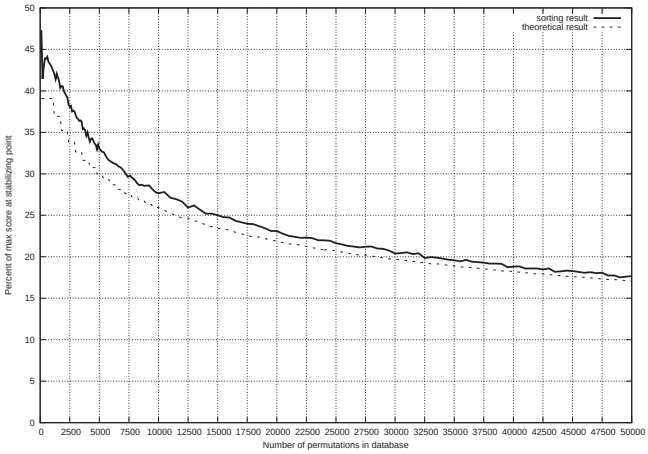


Fig. 6: Algorithm vs predicted equilibrium point, $n = 25$

7. Conclusion

We have seen that sorting, as a big data algorithm which uses queries to a big data set, is feasible. However, the success of the algorithm depends a great deal on the the scoring method which is used to evaluate progress towards the sorted goal. There are scoring methods, such as the cycle metric, which are very effective in producing the desired result, while some seemingly reasonable metrics have very poor results. Others, such as the inversion metric, have partial success. The size of the database as well as permutation size also have an effect on the sorting success, but clearly not to as great of a degree.

The sorting algorithm is just an abstract example of a broader class of big data search algorithms. From the results of the sorting algorithm, we can infer that the method of evaluating the data in the database is very significant in the effectiveness of the search. Furthermore, it is possible that the data set provides enough information to find the goal state, but due to the nature of the method used compare data elements against the data in the database, the result of the search may only be within a vicinity of the goal, or not at all. Thus, the effectiveness of big data still depends on the effectiveness of the particular higher-level tools used to analyze the data.

References

[1] Marco Chiani, Davide Dardari, and Marvin K. Simon. New exponential bounds and approximations for the computation of error probability in fading channels. *IEEE Transactions on Wireless Communications*, 2(4):840–845, July 2003.

[2] Vladimir Estivill-Castro and Derick Wood. A survey of adaptive sorting algorithms. *ACM Computing Surveys*, 24(4):441–476, December 1992.

[3] Ibrahim Abaker Targio Hashem, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, and Samee Ullah Khan. The rise of "big data" on cloud computing: Review and open research issues. *Information Systems*, 47:98–115, January 2015.

[4] James Hays and Alexei A. Efros. Im2gps: estimating geographic information from a single image. In *in IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[5] Sheldon Ross. *A First Course in Probability*, chapter 8. Macmillan College Publishing Company, New York, New York, fourth edition, 1994.

[6] Wenhorng Tian and Yong Zhao. *Optimized Cloud Resource Management and Scheduling*, chapter 2. Morgan Kaufmann. Elsevier Science & Technology Books, 2014.

Keyword Extraction by KNN considering Similarity among Features

Taeho Jo
Department of Computer and Information Engineering, Inha University, Incheon, South Korea

Abstract— *In this research, we propose that the K Nearest Neighbors should be used for extracting keywords from articles, considering the feature similarities. In the reality, the relations and the dependencies among features are available; the assumptions that the features are independent of each other violates against the reality. In this research, we define the similarity measure considering both feature values and features, interpret the keyword extraction into the binary classification where each word is classified into a keyword or non-keyword, and use it for modifying the K Nearest Neighbor. As the benefits from this research, we have the chance to represent words into the smaller dimensional numerical vectors and to improve the discriminations among vectors. Therefore, the goal of this research is to implement the keyword extraction systems with the benefits.*

Keywords: Keyword Extraction, Feature Similarity

1. Introduction

The keyword extraction refers to the process of extracting the essential words which reflect the entire content from an article. In this research, the keyword extraction is viewed into a binary classification where each word is classified into one of the two categories: 'keyword', or 'non-keyword'. We prepare sample words which are labeled with one of the two categories and encode them into their structured forms. By learning the sample words, we built the classification capacity, encode novice words into the structured forms, and classify them into one of the two categories. In this research, we use a supervised learning algorithm for the task which is viewed into the classification task.

Let us mention some problems which this research tries to solve. When discovering the dependencies among features of encoding texts or words into numerical vectors, the Bayesian networks was proposed as the approaches to the text mining tasks, but the complicated analysis is required for using them [1]. The assumption that features are independent of each other causes the requirement of many features for the robustness in encoding so. Since each feature has the very weak coverage, we cannot avoid the sparse distribution of each numerical vector where zero values are dominant with more than 95% [3]. Therefore, this research is intended to solve the problems by considering the feature similarity as well as the feature value similarity.

Let us mention what we propose in this research as its idea. In this research, we consider the both similarity measures, feature similarity and feature value similarity, for computing the similarity between numerical vectors. The keyword extraction is viewed into the binary classification where a supervised learning algorithm is applicable. The KNN (K Nearest Neighbor) is modified into the version which accommodates the both similarity measures and applied to the keyword extraction task. Therefore, the goal of this research is to improve the keyword extraction performance by solving the above problems.

We mention what we expect from this research as the benefits. Considering the both similarities which are covered in this research opens the potential way of reducing the dimensionality of numerical vectors for encoding texts. Computing the similarity between two texts by the two measures reflects more semantic similarity between words. It is expected to improve the discriminations among even sparse vectors by using the both kinds of similarities. Therefore, this research pursues the benefits for implementing the keyword extraction systems.

This article is organized into the four sections. In Section **??**, we survey the relevant previous works. In Section 3, we describe in detail what we propose in this research. In Section 4, we mention the remaining tasks for doing the further research.

2. Previous Works

Let us survey the previous cases of encoding texts into structured forms for using the machine learning algorithms to text mining tasks. The three main problems, huge dimensionality, sparse distribution, and poor transparency, have existed inherently in encoding them into numerical vectors. In previous works, various schemes of preprocessing texts have been proposed, in order to solve the problems. In this survey, we focus on the process of encoding texts into alternative structured forms to numerical vectors. In other words, this section is intended to explore previous works on solutions to the problems.

Let us mention the popularity of encoding texts into numerical vectors, and the proposal and the application of string kernels as the solution to the above problems. In 2002, Sebastiani presented the numerical vectors are the standard representations of texts in applying the machine learning algorithms to the text classifications [4]. In 2002, Lodhi et al. proposed the string kernel as a kernel function of raw texts in using the SVM (Support Vector Machine) to the text classification [5]. In 2004, Lesile et al. used the version of SVM which proposed by Lodhi et al. to the protein classification [6]. In 2004, Kate and Mooney used also the SVM version for classifying sentences by their meanings [7].

It was proposed that texts are encoded into tables instead of numerical vectors, as the solutions to the above problems. In

2008, Jo and Cho proposed the table matching algorithm as the approach to text classification [8]. In 2008, Jo applied also his proposed approach to the text clustering, as well as the text categorization [12]. In 2011, Jo described as the technique of automatic text classification in his patent document [10]. In 2015, Jo improved the table matching algorithm into its more stable version [11].

Previously, it was proposed that texts should be encoded into string vectors as other structured forms. In 2008, Jo modified the k means algorithm into the version which processes string vectors as the approach to the text clustering[12]. In 2010, Jo modified the two supervised learning algorithms, the KNN and the SVM, into the version as the improved approaches to the text classification [13]. In 2010, Jo proposed the unsupervised neural networks, called Neural Text Self Organizer, which receives the string vector as its input data [14]. In 2010, Jo applied the supervised neural networks, called Neural Text Categorizer, which gets a string vector as its input, as the approach to the text classification [15].

The above previous works proposed the string kernel as the kernel function of raw texts in the SVM, and tables and string vectors as representations of texts, in order to solve the problems. Because the string kernel takes very much computation time for computing their values, it was used for processing short strings or sentences rather than texts. In the previous works on encoding texts into tables, only table matching algorithm was proposed; there is no attempt to modify the machine algorithms into their table based version. In the previous works on encoding texts into string vectors, only frequency was considered for defining features of string vectors. Texts which are used as features of numerical vectors which represent words have their semantic similarities among them, so the similarities will be used for processing sparse numerical vectors, in this research.

3. Proposed Approach

This section is concerned with modifying the AHC (Agglomerative Hierarchical Clustering) algorithm into the version which considers the similarities among features as well as feature values, and it consists of the three sections. In Section 3.1, we describe the process of encoding words into numerical vectors. In Section 3.2, we do formally the proposed scheme of computing the similarity between two numerical vectors. In Section **??**, we mention the proposed version of AHC algorithm which considers the similarity among features as the approach to word clustering. Therefore, this article is intended to describe in detail the modified version of KNN algorithm and its application to the word clustering.

3.1 Word Encoding

This subsection is concerned with the process of encoding words into numerical vectors. Previously, texts each of which is consists of paragraphs were encoded into numerical vectors whose attributes are words. In this research, we attempt to encode words into numerical vectors whose attributes are text identifiers which include them. Encoding of words and texts into numerical vectors looks reverse to each other. In this Section, we describe in detail the process of mapping words into numerical vectors, instead of texts.

In the first step of word encoding, a word-document matrix is constructed automatically from a text collection called corpus. In the corpus, each text is indexed into a list of words. For each word, we compute and assign its weight which is called TF-IDF (Term Frequency-Inverse Document Frequency) weight [2], by equation (1),

$$w_i = TF_i(\log_2 N - \log_2 DF_i + 1) \quad (1)$$

where TF_i is the total frequency in the given text, DF_i is the total number of documents including the word, and N is the total number of documents in the corpus. The word-document matrix consists of TF-IDF weights as relations between a word and a document computed by equation (1). Note that the matrix is a very huge one which consists at least of several thousands of words and documents.

Let us consider the criterion of selecting text identifiers as features, given labeled sampled words and a text collection. We may set a portion of each text in the given sample words as a criteria for selecting features. We may use the total frequency of the sample words in each text as a selection criterion. However, in this research, we decided the total TF-IDF (Term Frequency and Inverse Document Frequency) which is computed by equation (1) as the criterion. We may combine more than two criteria with each other for selecting features.

Once some texts are selected as attributes, we need to consider the schemes of defining a value to each attribute. To each attribute, we may assign a binary value indicating whether the word present in the text which is given as the attribute, or not. We may use the relative frequency of the word in each text which is an attribute as a feature value. The weight of word to each attribute which is computed by equation (1) may be used as a feature value. Therefore, the attributes values of a numerical vector which represent a word are relationships between the word and the texts which are selected as features.

The feature selection and the feature value assignment for encoding words into numerical vectors depend strongly on the given corpus. When changing the corpus, different texts are selected by different values of the selection criterion as features. Even if same features are selected, different feature values are assigned. Only addition or deletion of texts in the given corpus may influence on the feature selection and the assignment of feature values. In order to avoid the dependency, we may consider the word net or the dictionary as alternatives to the corpus.

3.2 Feature Similarity

This subsection is concerned with the scheme of computing the similarity between numerical vectors as illustrated in Figure 1. In this research, we call the traditional similarity measures such as cosine similarity and Euclidean distance

feature value similarities where consider only feature values for computing it. In this research, we consider the feature similarity as well as the feature value similarity for computing it as the similarity measure which is specialized for text mining tasks. The numerical vectors which represent texts or words tend to be strongly sparse; only feature value similarity becomes easily fragile to the tendency. Therefore, in this subsection, as the solution to the problem, we describe the proposed scheme of computing the similarity between numerical vectors.

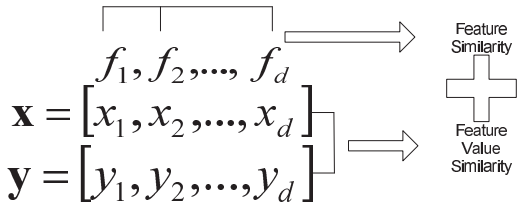


Fig. 1
THE COMBINATION OF FEATURE AND FEATURE VALUE SIMILARITY

Text identifiers are given as features for encoding words into numerical vectors. Texts are dependent on others rather than independent ones which are assumed in the traditional classifiers, especially in Naive Bayes [1]. Previously, various schemes of computing the semantic similarity between texts were developed [2]. We need to assign nonzero similarity between two numerical vectors where non-zero elements are given to different features with their high similarity. It is expected to improve the discriminations among sparse vectors by considering the similarity among features.

We may build the similarity matrix among features automatically from a corpus. From the corpus, we extract easily a list of text identifiers. We compute the similarity between two texts by equation (2),

$$s_{ij} = sim(d_i, d_j) = \frac{2 \times tf(d_i, d_j)}{tf(d_i) + tf(d_j)} \quad (2)$$

where $tf(d_i, d_j)$ is the number of words which are shared by both texts, d_i and d_j, and $tf(d_i)$ is the number of words which are included in the text, d_i. We build the similarity matrix which is consists of similarities between text identifiers given as features as follows:

$$S = \begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1d} \\ s_{21} & s_{22} & \cdots & s_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ s_{d1} & s_{d2} & \cdots & s_{dd} \end{pmatrix}.$$

The rows and columns in the above matrix,S, correspond to the d text identifiers which are selected as the features.

The texts, $d_1, d_2, ..., d_d$ are given as the features, and the two words, t_1 and t_2 are encoded into the two numerical vectors as follows:

$$t_1 = [w_{11}, w_{12}, ..., w_{1d}]$$

$$t_2 = [w_{21}, w_{22}, ..., w_{2d}].$$

The features,$d_1, d_2, ..., d_d$ are defined through the process which was described in Section 3.1. We construct the d by d matrix as the similarity matrix of features by the process mentioned above. The similarity between the two vectors are computed with the assumption of availability of the feature similarities, by equation (3),

$$sim(t_1, t_2) = \frac{\sum_{i=1}^{d} \sum_{j=1}^{d} s_{ij} w_{1i} w_{2j}}{d \cdot \|t_1\| \cdot \|t_2\|} \quad (3)$$

where $\|t_1\| = \sqrt{\sum_{i=1}^{d} w_{1i}^2}$ and $\|t_2\| = \sqrt{\sum_{i=1}^{d} w_{2i}^2}$. We get the value of s_{ij} by equation (2).

The proposed scheme of computing the similarity by equation (3) has the higher complexity as payment for obtaining the more discrimination among sparse vectors. Let us assume that two d dimensional numerical vectors are given as the input for computing the similarity between them. It takes only linear complexity, $O(d)$, to compute the cosine similarity as the traditional one. However, in the proposed scheme takes the quadratic complexity, $O(d^2)$. We may reduce the complexity by computing similarities of some pairs of features, instead of all.

3.3 Proposed Version of KNN

This section is concerned with the version of K Nearest Neighbor which considers both the feature similarity and the feature value one. The sample words are encoded into numerical vectors whose features are texts by the scheme which was described in section 3.1. The novice word is given as the classification target, and it is also encoded into a numerical vector. Its similarities with the sample words are computed by equation (3) for selecting nearest neighbors, in the proposed version. Therefore, in order to provide the detail algorithm, we describe the proposed KNN version, together with the traditional one.

The traditional KNN version is illustrated in Figure 2. The sample words which are labeled with the positive class or the negative class are encoded into numerical vectors. The similarities of the numerical vector which represents a novice word with those representing sample words are computed using the Euclidean distance or the cosine similarity. The k most similar sample words are selected as the k nearest neighbors and the label of the novice entity is decided by voting their labels. However, note that the traditional KNN version is very fragile in computing the similarity between very sparse numerical vectors.

The proposed KNN version is illustrated in Figure 3. Like the traditional version, a word is given as an input and it is encoded into a numerical vector. The similarities of the novice word with the sample ones are computed by equation (3) which was presented in section 3.2. Like the traditional version, k most similar samples are selected as the nearest neighbors, and the label of the novice is decided by voting their labels. The scheme of computing the similarity between

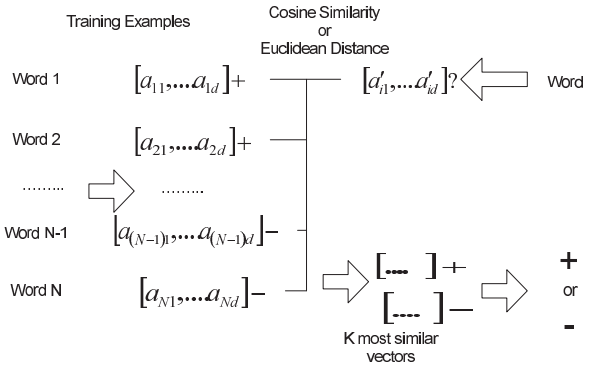


Fig. 2
THE TRADITIONAL VERSION OF KNN

numerical vectors is the essential difference between the two versions.

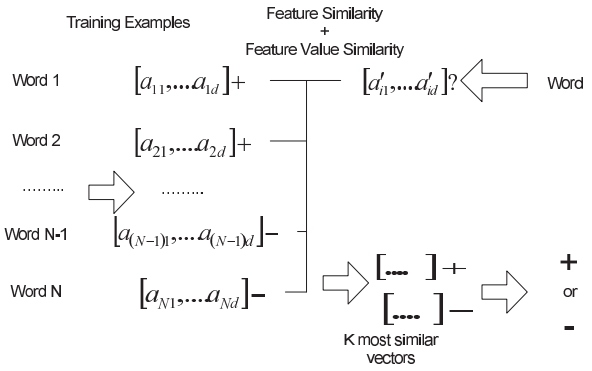


Fig. 3
THE PROPOSED VERSION OF KNN

We may derive some variants from the proposed KNN version. We may assign different weights to selected neighbors instead of identical ones: the highest weights to the first nearest neighbor and the lowest weight to the last one. Instead of a fixed number of nearest neighbors, we select any number of training examples within a hyper-sphere whose center is the given novice example as neighbors. The categorical scores are computed proportionally to similarities with training examples, instead of selecting nearest neighbors. We may also consider the variants where more than two variants are combined with each other.

Let us compare the both KNN versions with each other. In computing the similarity between two numerical vectors, the traditional version uses the Euclidean distance or cosine similarity mainly, whereas the proposed one uses the equation (3). Both versions are common in selecting k nearest neighbors and classifying a novice item by voting the labels of them. However, the proposed version is more tolerant to sparse numerical vectors in computing the similarities among them than the traditional version.

3.4 The Application to Keyword Extraction

This section is concerned with the scheme of applying the proposed KNN version which was described in section 3.3 to the keyword extraction task. Before doing so, we need to transform the task into one where machine learning algorithms are applicable as the flexible and adaptive models. We prepare the words which are labeled with 'keyword' or 'not' as the sample data. The words are encoded into numerical vectors by the scheme which was described in section 3.2. Therefore, in this section, we describe the process of extracting keywords from texts automatically using the proposed KNN with the view of the keyword extraction into a classification task.

In this research, the keyword extraction is viewed into a binary classification task, as shown in Figure 4. A text is given as the input, and a list of words is extracted by indexing the text. Each word is classified by the classifier into either of two labels: 'keyword' or 'not'. The words which are classified into 'keyword' are selected as the output of the keyword extraction system. For doing so, we need to collect words which are labeled with one of the two labels as sample examples, in advance.

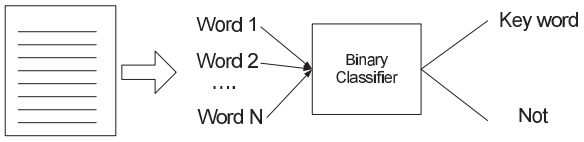


Fig. 4
MAPPING OF KEYWORD EXTRACTION INTO BINARY CLASSIFICATION

We need to prepare sample words which are labeled with 'keyword' or 'not', before classifying a novice one or ones. A text collection is segmented into sub-collections of content based similar words which are called domains, manually or automatically. We prepare sample words which are labeled manually, domain by domain. To each domain, we assign and train a classifier with the words in the corresponding sub-collection. When a text is given as the input, the classifier which corresponds to the most similar domain is selected among them.

We mention the process where an article is given as the input and a list of keywords is generated as the output. We nominate the classifier which corresponds to the sub-group which is similar as the given article, based on its content. A list of words is extracted by indexing the article, and each word is encoded into structured forms. The extracted words are classified by the nominated classifier into 'keyword' or 'not', and the words which are classified into the former are selected. The performance depends on the granularity of each sub-group; it should be optimized between the two factors: the amount of sample examples and the subgroup granularity.

Even if the keyword extraction is viewed into an instance of word categorization, it needs to be distinguished from the topic based word categorization. The word categorization is

given as a single multiple classification or multiple binary classifications, whereas the keyword extraction is fixed only to a single binary classification. In the word categorization, each word is classified semantically into one or some of the predefined topics, whereas in the keyword extraction, it is classified into an essential word, or not. In the word categorization, each word is classified by its meaning, whereas in the keyword extraction, it is classified by its relevancy to the given text. In the word categorization, when the given task is decomposed into binary classification tasks, a classifier is assigned to each topic, whereas, in the keyword extraction, a classifier is done to each domain.

4. Conclusion

Let us mention the remaining tasks for doing the further research. We need to validate the proposed approach in specific domains such as medicine, engineering, and economics, as well as in generic domains such as ones of news articles. We may consider the computation of similarities among some main features rather than among all features for reducing the computation time. We try to modify other machine learning algorithms such as Naive Bayes, Perceptrons, and SVM (Support Vector Machine) based on both kinds of similarities. By adopting the proposed approach, we may implement the word clustering system as a real program.

References

[1] T. Mitchell, *Machine Learning*, McGraw-Hill, 1997.
[2] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval: The Concepts and Technology behind Search*, Addison-Wesley, 2011.
[3] T. Jo, “The Implementation of Dynamic Document Organization using Text Categorization and Text Clustering" PhD Dissertation, University of Ottawa, Ottawa, Canada, 2006.
[4] F. Sebastiani, “Machine Learning in Automated Text Categorization", *ACM Computing Survey*, Vol. 34, pp. 1-47, 2002.
[5] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, “Text Classification with String Kernels", *Journal of Machine Learning Research*, Vol. 2, pp. 419-444, 2002.
[6] C. S. Leslie, E. Eskin, A. Cohen, J. Weston, and W. S. Noble, “Mismatch String Kernels for Discriminative Protein Classification", *Bioinformatics*, Vol. 20, pp. 467-476, 2004.
[7] R. J. Kate and R. J. Mooney, “Using String Kernels for Learning Semantic Parsers", in *Proc. ICCL ’06*, 2006, pp. 913-920.
[8] T. Jo and D. Cho, “Index based Approach for Text Categorization", *International Journal of Mathematics and Computers in Simulation*, Vol. 2, 2008, pp. 127-132.
[9] T. Jo, “Single Pass Algorithm for Text Clustering by Encoding Documents into Tables", *Journal of Korea Multimedia Society*, Vol. 11, 2008, pp. 1749-1757.
[10] T. Jo, “Device and Method for Categorizing Electronic Document Automatically", South Korean Patent 10-1071495, 2011.
[11] T. Jo, “Normalized Table Matching Algorithm as Approach to Text Categorization", *Soft Computing*, Vol. 19, 2015, pp. 849-849.
[12] T. Jo, “Inverted Index based Modified Version of K-Means Algorithm for Text Clustering", *Journal of Information Processing Systems*, Vol. 4, 2008, pp. 67-76.
[13] T. Jo, “Representationof Texts into String Vectors for Text Categorization", *Journal of Computing Science and Engineering*, Vol. 4, 2010, pp. 110-127.
[14] T. Jo, “NTSO (Neural Text Self Organizer): A New Neural Network for Text Clustering", *Journal of Network Technology*, Vol. 1, 2010, pp. 31-43.
[15] T. Jo, “NTC (Neural Text Categorizer): Neural Network for Text Categorization", *International Journal of Information Studies*, Vol 2, 2010, pp. 83-96.

A New Random Approach to Dimensionality Reduction

Augustine S. Nsang, A. Maikori, F. Oguntoyinbo and **H. Yusuf**

[1234]Department of Computer Science, School of Information Technology and Computing, American University of Nigeria, Yola By-Pass, PMB 2250, Yola, Nigeria

Abstract

In the field of machine learning, *dimensionality reduction* has become a very significant research area. If any information which is subject to analysis using machine learning techniques is represented by a large amount of information in high dimensional space, it is obvious that analyzing this information to get any results would be a very cumbersome process. It would be very helpful if we can reduce the dimensionality of this data set such that the result obtained by analyzing the reduced set is a good approximation to the result obtained by analyzing the original data set.

In this paper, we propose a random approach to reducing the dimensionality of a data set. We shall implement this approach and use it to reduce the dimensionality of two data sets. We shall also implement four already existing approaches to dimensionality reduction, and use them to reduce the same data sets. These methods shall include the *direct approach, principal component analysis, random projections,* and the *variance* approach. We shall then implement the *k-means* clustering approach and use it to separate the points in the two data sets mentioned here into *k* clusters (for some value of *k*). Finally, we shall compare our new random approach with the four existing approaches by the extent to which they preserve *k-means* clustering.

Keywords: Dimensionality reduction, k-means clustering, principal component analysis.

1. Introduction

Given a collection of *n* data points (vectors) in high dimensional space, it is often helpful to be able to project it into a lower dimensional space without suffering great distortion. In other words, it is helpful if we can embed a set of *n* points in *d*-dimensional space into a *k*-dimensional space, where $k << d$. This operation is known as *dimensionality reduction* (NR 2010a).

Dimensionality reduction has several advantages, the most important of which is the fact that with dimensionality reduction, we could drastically speed up the execution of an algorithm whose runtime depends exponentially on the dimensions of the working space (NR 2009a).

Dimensionality reduction can be applied in several domains, among which is the domain of clustering and classification. Clustering is the assignment of a set of observations into subsets (called clusters) so that observations in the same cluster are similar in some sense (Hartigan 1975; McKay 2003; S'winiarski & Pedrycz 1998). Clustering is a method of unsupervised learning, and a common technique for statistical data analysis used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics. Now, consider a data set, D, say, having 1000 observations in 5000 dimensions. Consider, a function, g, which assigns the observations in *D* into 25 different clusters. It would be helpful if we can reduce *D* to 400 dimensions, say, before applying *g* to it, speeding up the clustering process. Our goal here, thus, is to reduce *D* (with 5000 dimensions) to *D1* (with 400 dimensions) such that after applying *g* to both *D* and *D1* each observation in *D* should be assigned to the same cluster as the corresponding observation in *D1* (NR 2010b).

There are many known methods of dimensionality reduction. For some of these methods, each attribute in the reduced set is a linear combination of the attributes in the original data set. These include *random projections, principal component analysis* and several others (NR 2009a). Other dimensionality reduction methods, however, reduce a dataset to a subset of the original attribute set**.** These include the *Combined Approach (CA),* the *Direct Approach (DA),* and the *Variance* Approach (NR 2010a).

The rest of the paper is organized as follows. In Section 2, we shall discuss four existing dimensionality reduction techniques, which include *random projections, principal component analysis,* and the *variance* and *direct* approaches. In Section 3 we shall introduce our proposed random approach to dimensionality reduction. In Section 4 we shall discuss the *k-means* clustering approach for partitioning *n* data points into *k* clusters (for some value of *k*). In Section 5, we shall study the extents to which each of the five approaches implemented in this paper preserves *k*-means clustering and make comparisons with each other. Then we shall conclude this paper in Section 6.

2. Dimensionality Reduction Techniques

In this section, we shall examine four different existing reduction techniques. They include the following:

2.1 Random Projection

In *Random Projection*, the original *d*-dimensional data is projected to a *k*-dimensional ($k << d$) subspace through the origin, using a random $d \times k$ matrix *R* whose columns have unit lengths (Bingham E. 2001). If X_{nxd} is the original set of *n* *d*-dimensional observations, then

$$X^{RP}_{nxk} = X_{nxd} R_{dxk}$$

is the projection of the data onto a lower k-dimensional subspace (NR 2009b).

The key idea of random mapping arises from the Johnson Lindenstrauss lemma (Johnson W. B. 1984) which states that if points in a vector space are projected onto a randomly selected subspace of suitably high dimension, then the distances between the points are approximately preserved. In this paper, we shall implement a new approach that reduces a data set using this principle, but in which, unlike RP, the set of attributes in the reduced set is a proper subset of the set of attributes in the original set.

2.2 Principal Component Analysis (PCA)

Given *n* data points in $\Re^P$ as an *n x p* matrix *X*, we want to find the best *q*-dimensional approximation for the data *(q << p)*. The PCA approach achieves this by first computing the singular value decomposition of X. In other words, it finds matrices *U, D* and *V* such that $X = UDV^T$ where:

- *U* is an *n x n* orthogonal matrix (i.e. $U^TU = I_n$) whose columns are the left singular vectors of X;
- *V* is a *p x p* orthogonal matrix (i.e. $V^TV = I_p$) whose columns are the right singular vectors of X;
- *D* is an *n x p* diagonal matrix with diagonal elements $d_1 \geq d_2 \geq d_3 \ldots \geq d_p \geq 0$ which are the singular values of *X*. Note that the bottom rows of *D* are zero rows.
- Define U_q to be the matrix whose columns are unit vectors corresponding to the q largest left singular values of *X*. U_q is a *n x q* matrix.

The transformed matrix is given by (Bingham E. 2001):

$$X^{SVD} = X^T U_q$$

2.3 The Variance Approach

With the *Variance* approach, to reduce a dataset *D* to a data set D_R, we start with an empty set, *I,* and then add dimensions of *D* to this set in decreasing order of their variances (NR 2010a). That means that a set *I* of *r* dimensions will contain the dimensions of top r variances. Intuitively, it easy to justify why dimensions of low variance are left out as they would fail to discriminate between the data. (Indeed, in an extreme case where all the values along a dimension are equal, the variance is 0, and hence this dimension cannot distinguish between data points). Thus, let

$$I_r = \{i_1, \ldots, i_r\} \subset \{1, \ldots, n\},$$

the collection of dimensions corresponding to the top *r* variances. That is i_1 denotes the dimension of largest variance, i_2 the dimension of next larger variance, etc. The reduced data base is obtained by extracting the data corresponding to the selected dimensions. That is, project *D* on I_r to obtain:

$$D_R = D(:, I_r),$$

where D_R has the same number of rows as D and r columns: the i^{th} column of D_R is the column of the original database with the i^{th} largest variance.

2.4 The *Direct* Approach (NR 2010a)

To reduce a dataset D_{nxp} to a dataset containing *k* columns, the *Direct Approach* selects the combination of *k* attributes which best preserve the interpoint distances, and reduces the original dataset to a dataset containing only those *k* attributes. To do so, it first generates all possible combinations of *k* attributes from the original *p* attributes. Then, for each combination, *C,* it computes g_cm and g_cM given by:

$$g_cm = \min\{ \frac{\| f(u) - f(v) \|^2}{\| u - v \|^2} \}$$

$$g_cM = \max\{ \frac{\| f(u) - f(v) \|^2}{\| u - v \|^2} \}$$

where *u* and *v* are any two rows of *D,* and *f(u)* and *f(v)* are the corresponding rows in the dataset reduced to the attributes in C. The average distance preservation for this combination of attributes is then computed as:

$$g_cmid = (g_cm + g_cM)/2$$

In other words, with the *Direct Approach,* to find the average distance preservation for any combination of attributes, *C,* we reduce the original dataset directly to the dataset containing only the attributes in *C,* and then compute the average distance preservation for this combination using for the formulas above.

Thus, to reduce a dataset D_{nxp} to a dataset containing *k* columns, the *Direct Approach* reduces it to the dataset containing the combination of *k* attributes *C* with the highest g_cmid value.

3. Our New Random Approach

Our new random approach reduces a dataset containing *n* attributes to one containing *r* attributes by randomly selecting *r* attributes from the original *n* attributes. In other words, to reduce a data set *D* of dimensionality *d* to one of dimensionality *k,* using our proposed technique, a set S_k is formed consisting of *k* numbers selected at random from the set *S* given by:

$$S = \{x \in N \mid 1 \leq x \leq d\}$$

Then, our reduced set, D_R, will be given by:

$$D_R = D(:, S_k)$$

That is, D_R is a data set having the same number of rows as *D,* and if A_i is the i^{th} attribute of D_R, then A_i is the j^{th} attribute of *D* if *j* is the i^{th} element of S_k.

4. Application: Clustering High Dimensional Data (NR 2010a)

We will start by discussing *k-means* clustering of high dimensional data used to partition *n* data points into *k* clusters such that each observation belongs to the cluster with the nearest mean (Mac 67). We shall then compare our new

random approach with the other four dimensionality reduction approaches according to the degree to which they preserve - *k*-means clustering.

Given a set of data points ($x_1, x_2, \ldots, x_n$), $x_i \in \Re^d$, *k-means* partitions them into *k* sets, $S = \{S_i, i = 1, \ldots, k, k < n\}$, so as to minimize the within-cluster sum of squares. That is:

$$\text{argmin}_S \sum_{i=1}^{k} \sum_{x_j \in S_i} \| x_j - \mu_i \|^2$$

where μ_i is the mean of S_i.

4.1 Standard *k*-means algorithm

Given an initial set of *k* means $m_1^{(1)}, \ldots, m_k^{(1)}$ for the clusters S_i, $i = 1, 2, \ldots, k$ which may be specified randomly or by some heuristic, the algorithm proceeds by alternating between the following two steps:

1. **Assignment Step:** Assign each observation to the cluster with the closest mean:

$$S_i^{(t)} = \{x_j : | x_j - m_i^{(t)} \| \leq \| x_j - m_{i^*}^{(t)} \| \forall i^* = 1,\ldots,k\}$$

2. **Update Step:** Calculate the new means to be the centroid d of the observations in the cluster.

$$m_i^{(t+1)} = \frac{1}{| S_i^{(t)} |} \sum_{x_j \in S_i^{(t)}} x_j$$

The algorithm is deemed to have converged when the assignments no longer change.

5. K-Means Clustering Preservation by Each Approach

5.1 The Rand Index (NR 2010a)

Suppose we use *k-means* clustering to classify the *n* tuples of the data set *D* into *k* clusters, where $k \leq n$. Then we reduce *D* into *D1* using any of the four methods discussed above, and then use *k-means* clustering to classify the *n* tuples of *D1* into *k* clusters, where $k \leq n$. Our goal is to compare the extent to which *k-means* clustering of *D* is preserved in *D1* for each of the four approaches discussed above. The comparison is based on analyzing clustering performance. One approach for measuring clustering performance is the *rand* index.

The *rand* index evaluates clustering based on their complete agreement, that is based on those pairs of data that are clustered **together** by each clustering and those which are clustered in **different clusters** by each clustering. Let *D* and *D1* be the original and reduced data sets. For each $u \in D$ we denote by *f(u)* the corresponding data point in *D1*. Let *S* and *S1* denote the collection of clusters obtained from *D* and *D1* respectively by applying *k-means* clustering. Define the following:

- *a:* the number of pairs of rows *(u, v)* of *S* such that *u* and *v* belong to the **same cluster** in *S* and the corresponding rows *f(u)* and *f(v)* belong to the **same cluster** in *S1*
- *b:* the number of pairs of rows *(u, v)* of *S* such that *u* and *v* belong to **different clusters** in *S* and the corresponding rows *f(u)* and *f(v)* belong to **different clusters** in *S1*
- Define $n_p = a + b$. That is, n_p measures the total number of pairs for which the clustering results agree in the original and reduced space
- The *rand* index is defined as:

$$rand = 100x \frac{n_p}{| D | (| D | -1)/2}$$

where |D| is the number of elements in *D*. Thus the *rand* index is the percentage of pairs in which clustering results in the original and reduced space agree, or, the extent (computed as a percentage) to which the clustering of *D* is preserved in *D1*.

Example:

Consider the following dataset, *D*, given in *Table 1* below:

10	1	100	5	80	70	5	12
25	2	150	9	140	65	8	15
35	3	107	15	180	40	10	4
15	5	6	13	90	25	15	7
80	8	9	11	105	5	18	10
2	12	14	16	120	15	16	16
18	18	20	7	147	30	30	23

Table 1: The *D* Data Set

Reducing the dataset *D* from 8 columns to 5 columns using the *Variance* Approach gives us the data set D_v given in *Table 2* below:

10	100	80	70	5
25	150	140	65	8
35	107	180	40	10
15	6	90	25	15
80	9	105	5	18
2	14	120	15	16
18	20	147	30	30
10	100	80	70	5
25	150	140	65	8
35	107	180	40	10
15	6	90	25	15
80	9	105	5	18
2	14	120	15	16
18	20	147	30	30

Table 2: The D_v Data Set

Now, applying *k-means* clustering on the dataset *D* (with $k = 4$) gives us the results in the dataset *D-cl* (*Table 3)* below. In this table, the last entry of each row is the cluster to which that row has been assigned by the *k-means* algorithm.

10	1	100	5	80	70	5	12	1
25	2	150	9	140	65	8	15	1
35	3	107	15	180	40	10	4	2
15	5	6	13	90	25	15	7	3
80	8	9	11	105	5	18	10	3
2	*12*	*14*	*16*	*120*	*15*	*16*	*16*	*4*
18	18	20	7	147	30	30	23	4

Table 3: The *D-cl* Data Set

Next, applying *k-means* clustering on the dataset D_v (with k = 4) gives us the results in the dataset D_v-*cl* (*Table 4)* below.

10	100	80	70	5	1
25	150	140	65	8	1
35	107	180	40	10	2
15	6	90	25	15	3
80	9	105	5	18	3
2	14	120	15	16	4
18	20	147	30	30	4

Table 4: The D_v-*cl* Data Set

Thus, in this case, a = 3 and b = 18. Thus n_p = 21.

The total number of pairs of tuples is given by:

totnpairs = (7 * 6)/2 = 21

Thus the degree to which the *k-means* clustering of the original dataset is preserved by the *variance* approach is given by:

rand = $(n_p/\text{totnpairs})*100$ **%**

= 100%

5.2 Preservation of *k-means* clustering by each of the five approaches

The results in *Tables 5* and *6* were obtained when the datasets D_w and D_y (*Tables 7 and 8* respectively*)* were reduced using each of the five approaches from **10** attributes to **3, 4, 5, 6, 7, 8** and **9** attributes respectively, and the *rand* index used to determine the degree to which the *k-means* clustering of these two datasets were preserved by each reduction. D_w was extracted from the *Wine Quality* data set on *UCI machine learning repository* [UCI], and modified a little to work with the dimensionality reduction techniques used in this paper. To modify this data set, a program was written which checks if two rows of the data set have a common entry within a given column, and adds one to the second entry. D_y was extracted from the *Arrythmia* data set on *UCI machine learning repository* [UCI], and was modified the same as D_w.

Method	Extent to which K-Means Clustering is Preserved						
	3Att	**4Att**	**5Att**	**6Att**	**7Att**	**8Att**	**9Att**
RP	78%	82%	75.7%	70%	79.2%	76%	79%
PCA	76%	71%	75.8%	76.5%	75.8%	76%	77%
Var	78%	90%	91.7%	90%	90%	96%	100%
DA	78%	90%	91.7%	90%	90%	76%	100%
NRA	84%	85%	82.8%	81%	78%	79%	81%

Table 5: Comparing the DA, PCA, RP, Var and NRA Approaches for *k-means* clustering preservation (with k = 7) using the dataset D_w, with *rand* as the clustering performance measure.

Method	Extent to which K-Means Clustering is Preserved						
	3Att	**4Att**	**5Att**	**6Att**	**7Att**	**8Att**	**9Att**
RP	81%	84%	80.9%	84%	84.4%	84%	81.3%
PCA	86%	85%	82.3%	84%	83.6%	90%	85.3%
Var	93%	94%	96.6%	97%	97.8%	100%	100%
DA	72%	89%	89.2%	97%	95.7%	100%	99%
NRA	85%	79%	86.4%	84%	85.2%	85%	83.7%

Table 6: Comparing the DA, PCA, RP, Var and NRA Approaches for *k-means* clustering preservation (with k = 7) using the dataset D_y, with *rand* as the clustering performance measure.

The above results show that:

- The *Variance* Approach does better than all the other approaches in preserving the *k-means* clustering of the original datasets (on the average)
- The *Direct Approach* is almost as good as the *Variance* Approach in preserving the *k-means* clustering of the original datasets
- The *New Random Approach* does slightly better than the *RP* and *PCA* Approaches in preserving the *k-means* clustering of the original datasets
- The *RP* and *PCA* Approaches are approximately as good as each other in preserving the *k-means* clustering of the original datasets

7	27	36	27	45	45	17	11	3	5
63	3	34	16	49	14	132	4	33	49
81	28	4	69	5	3	97	51	326	44
72	23	32	85	58	47	186	56	319	40
79	30	39	92	65	54	193	63	333	47
88	35	11	76	12	10	104	58	340	51
62	32	16	7	52	17	136	49	318	54
14	34	43	34	59	52	24	18	10	52
70	10	41	23	56	21	139	25	40	56
95	22	50	15	44	28	129	38	322	59
102	41	48	145	33	11	63	8	2	63
86	37	18	42	35	24	109	47	314	53
93	18	37	12	4	16	75	2	325	70
66	16	25	22	51	48	143	12	354	66
83	42	62	1925	11	41	172	102	298	67
73	17	38	29	32	35	112	14	332	55
77	48	46	11	46	31	99	28	324	36
69	66	55	19	29	29	82	9892	347	39
74	55	42	18	40	38	171	17	312	60
65	31	14	75	72	34	133	55	329	50
76	73	69	26	36	36	89	9899	361	46
64	38	45	36	38	19	102	19	317	35
68	26	49	17	63	55	122	3	368	48
90	67	21	43	74	25	168	37	305	58
80	62	76	13	66	23	142	65	342	61
21	25	53	9	53	56	245	62	339	57
97	24	35	1	73	42	146	10	345	65
28	49	60	87	79	32	141	61	338	74
109	69	83	25	47	59	153	21	375	77
100	39	57	2	54	37	114	6	31	71
85	45	67	14	86	2	149	74	32	81
104	14	90	32	42	7	47	34	382	68
116	46	64	205	19	66	1	9	389	72
111	12	97	50	80	43	117	39	349	79
58	76	2	1495	93	22	179	69	337	37
87	56	71	24	8	49	123	5	396	42
107	53	23	54	100	39	156	41	331	84
35	33	74	33	60	73	138	13	313	28
94	52	81	1795	57	80	106	149	321	43
101	59	88	1802	64	87	113	156	328	64
67	44	95	39	172	63	158	44	311	78
123	80	102	57	173	70	157	72	38	34
42	87	26	74	69	77	16	54	320	88
108	94	27	21	71	33	152	48	336	91
130	51	33	28	6	40	154	76	352	98
137	60	31	35	107	94	167	31	316	95
118	101	40	44	70	84	206	32	327	73
125	74	109	51	87	62	207	46	346	80
49	108	47	81	76	91	23	68	334	102
132	19	56	5	67	46	15	26	343	62
114	40	52	30	62	101	173	24	335	86
115	81	29	46	68	53	124	35	341	93
139	58	54	53	78	98	96	1	353	109
71	88	63	8	94	60	108	53	359	105
75	2	59	37	147	108	160	45	403	38
146	95	70	20	39	50	191	15	348	87
121	21	104	119	43	44	213	83	309	85
6	47	61	124	48	5	147	79	47	92
122	115	15	60	114	67	78	52	360	94
144	9	78	40	101	51	111	33	350	41
82	29	24	49	92	115	130	86	303	116
13	54	68	131	55	12	161	93	54	99
56	61	7	67	77	122	151	22	302	69
129	122	22	88	121	74	85	59	367	101
128	129	111	56	108	129	150	27	315	123
153	63	3	47	85	57	12	90	357	75
78	68	66	64	18	27	131	66	364	112
151	102	13	61	41	81	174	16	356	119
158	65	73	63	81	136	103	40	410	106
165	136	85	95	26	105	105	97	355	126

.

Table 7: The D_w Data Set

75	0	190	80	91	193	371	174	121	-16
56	1	165	64	81	174	401	149	39	25
54	7	172	95	138	163	386	185	102	96
55	14	175	94	100	202	380	179	143	28
82	21	197	87	88	181	360	177	103	-9
13	28	169	51	107	167	321	181	91	107
40	8	160	52	77	129	377	133	77	77
49	15	162	54	78	0	376	157	70	67
44	35	168	56	84	118	354	160	63	61
50	22	167	67	89	130	383	156	73	85
62	42	170	72	102	135	408	163	83	72
45	29	179	86	98	143	373	150	65	12
61	36	186	58	85	155	382	170	81	-24
30	49	177	73	105	180	355	164	104	68
51	43	174	88	112	158	399	184	94	46
47	50	150	48	75	132	350	169	72	36
68	56	171	59	82	145	347	176	61	84
46	57	158	65	70	120	353	122	52	57
73	63	193	63	119	154	392	175	90	73
57	64	166	79	96	188	406	158	79	-12
28	71	181	93	83	251	390	189	183	50
52	70	176	74	90	122	336	191	78	81
36	78	153	75	71	139	364	183	82	62
64	85	200	66	103	157	413	143	92	4
89	92	188	55	110	140	388	198	89	52
58	77	183	101	109	128	389	195	60	-34
34	84	184	108	94	186	387	224	125	90
31	99	195	61	95	161	407	168	97	10
63	106	164	100	97	164	420	381	99	-8
65	113	202	83	117	147	400	301	96	-37
53	91	182	85	92	171	415	172	98	-52
72	120	163	68	99	136	339	152	76	13
71	127	209	115	124	125	367	190	84	38
59	134	155	70	106	137	368	148	111	9
69	98	204	82	131	152	357	129	101	49
79	141	216	45	69	178	378	137	80	69
78	105	207	122	145	142	366	161	108	54
35	148	178	129	113	200	385	188	74	48
76	155	191	60	80	185	361	166	107	-2
66	112	189	136	101	170	422	159	106	-57
43	162	157	71	87	162	397	141	105	53
96	169	223	89	79	168	427	167	115	74
37	176	230	62	120	175	346	165	110	75
41	183	214	96	126	159	404	144	71	59
103	190	237	107	86	177	374	147	113	-18
110	197	156	69	73	166	434	138	120	-11
83	204	244	78	152	156	322	186	112	18
86	211	221	57	108	144	369	171	119	30
48	218	159	76	127	228	429	130	122	63
90	225	173	103	115	149	379	205	118	-14
117	232	198	53	116	187	393	151	127	1
42	239	205	143	104	184	359	197	87	34
39	119	228	90	159	182	356	162	129	11
24	246	212	81	134	192	370	142	68	64
80	126	235	150	123	209	411	113	132	58
93	253	251	77	133	191	441	202	88	37
38	260	258	84	93	7	418	193	34	14
70	133	196	114	111	173	448	182	86	-4
60	140	242	157	140	176	372	196	93	42
32	267	265	121	118	150	414	209	54	88
1	147	110	10	122	121	287	212	67	126
77	274	272	128	141	189	324	155	75	40
85	281	249	135	76	199	394	173	136	95
27	288	211	102	166	196	381	216	85	66
97	295	218	97	130	382	209	63	117	60
100	302	225	142	129	117	363	219	100	56
124	309	279	9	173	165	410	204	150	32
26	316	286	149	148	206	455	200	95	80
87	323	185	92	136	203	436	207	109	71
92	330	219	110	143	213	443	180	116	-36

Table 8: The D_y Data Set

6 Conclusion

We have introduced a new approach to dimensionality reduction which reduces a dataset containing *n* attributes to one containing *r* attributes by randomly selecting *r* attributes from the original *n* attributes. Apart from this *new random approach*, we implemented four existing approaches to dimensionality reduction which include *random projection*, *principal component analysis*, the *variance approach* and the *direct approach.*

We also used the *k-means* clustering approach for partitioning *n* data points into *k* clusters (for some value of *k*). Then we compared our new random approach with the four existing approaches for *k-means* clustering preservation. We realized that our *New Random Approach* does slightly better than the *PCA* and *RP* Approaches in preserving the *k-means* clustering of the original dataset. We also realized that the *Variance* Approach performs better than every other approach in preserving the *k-means* clustering of the original datasets, while *RP* and *PCA* are approximately as good as each other in preserving the *k-means* clustering of the original datasets.

References

Nsang, A. 2011. *Novel Approaches to Dimensionality Reduction and Applications, An Empirical Study*. Lambert Academic Publishing.

Nsang A., Ralescu A. 2010, Approaches to Dimensionality Reduction to a Subset of the Original Dimensions. In *Proceedings of the Twenty-First Midwest Artificial Intelligence and Cognitive Science Conference (MAICS 2010), South Bend, IN.,* 70 - 77.

Nsang A., Ralescu A. 2009, A Review of Dimensionality Reduction and Their Applications. In *Proceedings of the Twentieth Midwest Artificial Intelligence and Cognitive Science Conference (MAICS 2009), Fort Wayne, IN.,* 118 - 123.

Hartigan, J. A. 1975. *Clustering Algorithms.* Wiley. MR0405726. ISBN 0-471-35645-X.
MacKay, D. 2003. *Information Theory, Inference and Learning Algorithms.* Cambridge University Press.

S'winiarski, R.; Cios, K., and Pedrycz, W. 1998. *Data Mining Methods for Knowledge Discovery.* Kluwer Academic. ISBN 0-7923-8252-8.

Kotsiantis, S. 2007. Supervised Machine Learning: A Review of Classification Techniques, 249-268.

Nsang A., Ralescu A. 2010. More Dimensionality Reduction to a Subset of the Original Attribute Set. In *Proceedings of the Twenty-First Midwest Artificial Intelligence and Cognitive Science Conference (MAICS 2010), South Bend, IN.,* 109-116.

Bingham E., Manilla H. 2001, Random Projections in Dimensionality Reduction: Applications to Image and Text Data. In *Conference on Knowledge Discovery Discovery in Data, Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 245-250.

Nsang A., Ralescu A. 2009, Query-Based Dimensionality Reduction Applied to Text and Web Data. In *Proceedings of the Twentieth Midwest Artificial Intelligence and Cognitive Science Conference (MAICS 2009), Fort Wayne, IN.,* 129 – 136.

Johnson W. B., Lindenstrauss J., 1984. Extensions of Lipshitz mapping into Hilbert Space. *Contemporary Mathematics.*

MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. In *Proc. Fifth Berkeley Symposium On Mathematics, Statistics and Probability, Volume 1 (University of California Press), 281 – 297.*

http://archive.ics.uci.edu/ml/datasets.html

Multi-Label classification for Mining Big Data

Passent M. El-Kafrawy
Math and Computer Science Dept.
Faculty of Science, Menofia University
Shebin Elkom, Egypt
basant.elkafrawi@science.menofia.edu.eg

Amr M. Sauber
Math and Computer Science Dept.
Faculty of Science, Menofia University
Shebin Elkom, Egypt
amrmausad@computalityit.com

Awad Khalil
Dept. of Computer Science and Engineering, American University in Cairo Egypt
akhalil@aucegypt.edu

***Abstract*—In big data problems mining requires special handling of the problem under investigation to achieve accuracy and speed on the same time. In this research we investigate the multi-label classification problems for better accuracy in a timely fashion. Label dependencies are the biggest influencing factor on performance, directly and indirectly, and is a distinguishing factor for multi-label from multi-class problems. The key objective in multi-label learning is to exploit this dependency effectively. Most of the current research ignore the correlation between labels or develop complex algorithms that don't scale efficiently with large datasets. Hence, the goal of our research is to propose a fundamental solution through which preliminary identification of dependencies and correlations between labels is explicit from large multi-label datasets. This is to be done before any classifiers are induced by using an association rule mining algorithm. Then the dependencies discovered in the previous step are used to divide the problem into subsets depending on the correlation between labels for parallel classification. The experimental results were evaluated using Accuracy, Hamming Loss, Micro-F-Measure and Subset Accuracy on a variety of datasets. The proposed model exploits all correlations among labels in multi-label datasets easily, facilitating the process of multi-label classification, increasing accuracy and performance as time was decreased while achieving higher accuracy.**

***Keywords*-Multi-Label classification; data mining; big data analytics;**

I. Introduction

Data is acquired nowadays in huge amounts from different resources in a fast paste. These huge amounts of data are considered the main asset of the business, however, the use of such big data is gained when knowledge could be deduced intelligently and in a timely fashion. Current research seeks to exploit new techniques to handle such amount of data and explores ways to analytically find patterns in the data to deduce knowledge to add higher values to the business intelligence ecosystems. One of the major concerns in most classification problems is the existence of multi-label in multi-class problems. When each item could be associated with multiple labels of a certain classification system, this is known as multi-label classification.

A multitude of other sources have also, knowingly or not, embraced the multi-label context. Domains such as microbiology or medicine often inherently require a multi-label scheme: a single gene may influence the production of more than one protein, and a patients symptoms may be linked to multiple ailments. This explains the explosion of interest in multi-label classification in the academic literature over recent years.

Tsoumakas and Katakis [1], categorized multi-label classification methods into two main categories: problem transformation and algorithm adaptation. In problem transformation a multi-label problem is transformed into one or more single-label problems. This scheme uses common off-the-shelf single-label classifiers and thus avoids the restrictions of a certain classification paradigm. This is opposed to algorithm adaptation, where a specific classifier is modified to carry out multi-label classification; often highly suited to specific domains or contexts but not as flexible and has a high computational complexity.

Recently Dembczynski [2], provided a clarification and formalization of label dependence in multi-label classifications. According to them, one must distinguish between unconditional and conditional label dependence. Modeling unconditional dependencies is good enough for solving multi-label classification problems and improves performance; in contrast modelling conditional dependencies does not improve the predictive performance of the classifier as such [3].

Exploiting label dependence becomes a popular motivation in recent research, multi-label learning methods have been developed [4] for further research. Usually, label correlations are given beforehand or can be derived directly from data samples by counting their label co-occurrences. However, in some problem domains unfortunately this type of knowledge is unavailable, therefore new techniques are required to obtain this valuate knowledge. Accordingly, the goal of our research is to propose a fundamental solution through which preliminary identification of dependencies and correlations between labels explicitly from multi-label datasets for big data. This is to be done before any classifiers are induced by using an association rule mining algorithm. Then the dependencies discovered in the previous step are used to construct a multi-label classifier. The problem is such divided into subproblems for *parallel* classification that improves execution time and accuracy of classification. This leads to exploit all dependencies between labels easily and

in a timely fashion to facilitate the multi-label classification process. Consequently, this would allow us to achieve increased efficiency, accuracy, speed which is the main goal of our research.

This paper is organized as follows. Section II defines the problem under investigation and presents the related work. Section III explains our proposed method; aimed at improving both performance and accuracy of multi-label classifiers. The experimental results are illustrated and discussed in Section IV. Finally, Section V summarizes the conclusion and the future work.

II. Problem Definition

Before defining the problem, related concepts and notations are introduced. The basic definition of multi-label systems need to be clarified, in order to clearly formulate the main problem under investigation. After which recent related work is presented.

A. Multi-label Classification

Our learning model is the following standard extension of the binary case. Assume that an instance $x \in X$ can be associated with a subset of labels y, which is referred to as the relevant set of labels for x, for which we use subset notation $y \subseteq [L]$ or vector notation $y \in [0,1]^L$, as dictated by convenience. Assume Y is a given set of predefined binary labels $Y = \lambda_1, .., \lambda_L$. For a given set of labeled examples $D = x_1, x_2, ..., x_n$ the goal of the learning process is to find a classifier $h : X \rightarrow Y$, which maps an object $x \in X$ to a set of its classification labels $y \in Y$, such that $h(x) \subseteq \lambda_1, ..., \lambda_L$ for all x in X.

The main feature distinguishing multi-label classification from a regular classification task is that a number of labels have to be predicted simultaneously. Thus, exploiting potential dependencies between labels is important and may improve classifier predictive performance. In problem transformation, a multi-label problem is transformed into one or more single-label problems, single label classification problems are solved with a commonly used single-label classification approach and the output is transformed back into multi-label representation. In this category, the common methods used are the label power-set [1], [5], binary relevance [1] and pair-wise methods [6], [7].

There are two significant methods in problem transformation methods: the pruned sets method [8], and extended in [8], and the classifier chains method [9], and ensemble schemes for both methods. Both these methods put a heavy emphasis on efficiency, achieve better predictive performance than other methods and can scale to large datasets. For multi-label dataset with a large number of labels a Hierarchy Of Multi-label Classifiers (HOMER) [10] method was proposed, and an ensemble method called RAKEL [5].

B. Related Work

Few works on multi-label learning have directly discovered existing dependencies among labels in advance before any classifiers are induced.

In [3], the author proposed the ConDep and ChiDep algorithms, which explicitly identify conditional and unconditional dependent label pairs between labels from a training set. The proposed method is based on analyzing the number of instances in each category. They apply the chi-square test for independence to the number of instances for each possible combination of two categories. The level of dependence between each two labels in the dataset is thus identified. Then decompose the original set of labels into several subsets of dependent labels, build an LP classifier for each subset, and then combine them as in the BR method. The results confirm that modeling unconditional dependencies is good enough for solving multi-label classification problems. The disadvantages of ConDep and ChiDep algorithms are complexity and needs improvement in ChiDep time performance.

In another rresearch [11], heterogeneous information networks are used to facilitate the multi-label classification process by mining the correlations among labels from heterogeneous information networks. They proposed a novel solution, called PIPL, to assign a set of candidate labels to a group of related instances in the networks. Unlike previous work, the proposed PIPL can exploit various types of dependencies among both instances and labels based upon different meta-paths in heterogeneous information networks. Empirical studies on real-world tasks effectively boost classification performances; but only tested on a bioinformatics datasets, which are a heterogeneous network.

III. Proposed Algorithm

We shall study the multi-label learning process by mining the correlations among labels from association rules. This is achieved by exploiting the correlation between labels and divide the dataset into two subsets: the items with unconditional dependencies and the ones with conditional dependencies. Each subset is feed in a parallel framework in the proposed algorithm for further speed up of the classification process.

A. Discovering dependences between labels

There are many multi-label learning methods to handle multi-label datasets and all these methods deal with the relations between labels in a different way. These relations between labels are necessary to facilitate the learning process. However, disparity on the degree of correlations between the labels may exists, where some labels may have high correlations while others may have medium correlations, and some other labels may not have any correlations among them. Multi-label learning methods might handle labels either by ignoring the correlations completely (this

doesnt allow us to exploit the dependencies between labels) or take Holistic correlations between all (these increases the complexity of the learning process). To overcome these problems we shall propose a solution that is based on the idea that every problem is treated as a unique case where we explore each set of labels as a separate case.

First of all we shall discover correlations between labels, which is achieved by using association rules. Association rules [12] mining is an interesting branch of data mining that focuses on looking at the discovery of associations and correlations among items in large transactional or relational data sets. It provides knowledge about the underlying system to a set of data and can be interpreted as implications, so that the presence of certain elements implies the occurrence of others.

$$Y = Y' \cup Y'', |Y'| \geq 0, |Y''| \geq 0 \quad (1)$$

Y all label, Y' uncorrelated label, Y'' correlated label.

The Second step, we shall use divide and conquer technique to divide the datasets into sub datasets, this depends on the degree of correlations among labels. Basically we might divide the datasets into two datasets. One that has correlated labels and the other that has uncorrelated labels

The third step, we apply a multi-label classifier on each data-subset considering each subset as a distinct problem, solved by itself.

Finally, integrating the results of all data-subsets is required by combining the results using average function. A pseudo-code for the proposed method, see algorithm III-B

B. Exploiting unconditional dependencies

Exploiting conditional dependencies is done through applying a priori algorithm on labels in multi-label dataset. We shall obtain all associations and correlations among labels that satisfy minimum support count, which is the number of transactions that contain the labels. Accordingly, labels that are correlated are pointed out and the rest of the dataset are items with uncorrelated labels.

The rule $A \Rightarrow B$ holds in the transaction set D with support s, where s is the percentage of transactions in D that contain $A \cup B$. This is taken to be the probability, $P(A \cup B)$. The rule $A \Rightarrow B$ has confidence c in the transaction set D, where c is the percentage of transactions in D containing A that also contain B. This is taken to be the conditional probability, $P(B \cup A)$. That is,

$$Support(A \Rightarrow B) = P(A \cup B) \quad (2)$$

$$Confidence(A \Rightarrow B) = P(B|A) = \frac{support(A \cup B)}{support(A)} \quad (3)$$

Rules that satisfy both a minimum support threshold (min-sup) and a minimum confidence threshold (min-conf) are called strong. Strong rules are used to define items that correlate. Thus the problem of mining association rules can be reduced to that of mining frequent itemsets. In general, association rule mining can be viewed as a two-step process:

1) Find all frequent itemsets: by applying apriori algorithm on labels, each of these itemsets will occur at least as frequently as a predetermined minimum support count, min-sup.
2) Generate strong association rules from the frequent itemsets: satisfing minimum support and minimum confidence.

Algorithm: A pseudo-code for the proposed method.

```
P=A priorie (L)  # return related label
C= []
For (i=0; i<= P. length; i++)
    C = C U L[P[i]];
Un_C = L-C
Multi thread (
    Result1 = classifier (D + UN_C)
    Result2 = classifier (D + C) )
Output = Avg (result1+result2)
```

IV. Experimentation of the Proposed System

A. Evaluation Metrics

In our experiments, we used various evaluation measures that have been suggested by Tsoumakas et.al. [13]. We chose four popular measures, multi-label example-based classification accuracy, hamming loss, subset accuracy and label-based micro-averaged F-measure. In the definitions below, y_i denotes the set of true labels of example x_i and $h(x_i)$ denotes the set of predicted labels for the same examples.

1) Hamming loss: evaluates how many times an example-label pair is misclassified. The performance is perfect when hamming-loss (h) = 0.

$$Hamming - loss(h) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{Q}|h(x_i)\delta y_i| \quad (4)$$

where δ stands for the symmetric difference between two sets, N is the number of examples and Q is the total number of possible class labels.

2) Accuracy: computes the percentage of correctly predicted labels among all predicted labels. Accuracy is averaged over all dataset examples as follows:

$$Accuracy(h) = \frac{1}{N}\sum_{i=1}^{N}|\frac{h(x_i) \cap y_i}{h(x_i) \cup y_i}| \quad (5)$$

3) Subset Accuracy: or classification accuracy is defined as follows:

$$Subset - accuracy(h) = \frac{1}{N}\sum_{i=1}^{N} I(h(x_i) = y_i) \quad (6)$$

where I(true)=1 and I(false)=0.

Table I
DATASETS USED IN THE EXPERIMENT: INFORMATION AND STATISTICS

Dataset	Domain	Instance	Attribute	Label	L_{card}
Scene	image	2407	294	6	1.074
Yeast	biology	2417	72	14	4.237
Birds	audio	645	260	19	1.014
Slashdot	text	3782	1079	22	1.18
Genbase	biology	662	1186	27	1.252
Medical	text	978	1449	45	1.245
Enron	text	1702	1001	53	3.378
Cal500	music	502	68	174	26

4) Micro F-measure: is the harmonic mean between micro-precision and micor-recall. MicroF1 is defined as:

$$MicroF1 = \frac{2 * MicroPrecision * MicroRecall}{MicroPrecision + MicroRecall} \quad (7)$$

B. Datasets

Multi-label classification problems appear in a wide range of real world situations and applications. The datasets that are included in the experimental setups cover three main application areas in which multi-label data is frequently observed: text categorization, multimedia classification and bioinformatics. All datasets were mainly retrieved from the repository of the Mulan Java Library [14]; as summarized in Table I.

C. Procedure

We used a priori algorithm to find frequent itemsets. The proposed method has been implemented in Java and integrated with Weka [15] and Mulan [14] open-source Java libraries. We applied multi-label classification using the proposed method on the standard multi-label classification methods such as (Binary Relevance (BR), Label Powerset (LP), Classifier Chains (CC), Pruned Sets (PS), CalibratedLabelRanking (CLR), and HOMER). Also, applied on some ensemble methods: (Random k-labelsets (RAKEL), Ensemble of Classifier Chains (ECC) and Ensemble of Pruned Sets (EPS)).

The results of applying the proposed model using different multi-label classification methods on given datasets are compared to traditional multi-label learning without dividing the multi-label datasets. After that, the proposed model with the best performing multi-label classification is compared to ChiDep [7].

The experiments were conducted using the 10-fold cross-validation methodology. All the algorithms were supplied with Wekas J48 implementation of a C4.5 tree classifier as a single-label base learner.

D. Parameter configuration

The execution of the a priori algorithm was performed with the following parameters: support was set to 0.01.

As for all participating classification algorithms, all configurable parameters were set to their optimal values as reported in the relevant papers. BR, LP and CC do not require parameters. For HOMER, its balanced k means version with k = 3 was used. The PS methods required two parameters p (set to 2) and strategy (set to 2) for each dataset, as proposed by [8].

The number of models in the ECC methods was set to 10 as proposed by [9], while for RAKAL the number of models was set to 10 for all datasets and the size of the label-sets K for each dataset was set to half the number of labels. For EPS, at each dataset p and strategy parameters were set to the same values as those used for the PS method. EPS requires additional parameter, the number of models was set to 10. For all ensemble methods the majority voting threshold was set to 0.5.

V. RESULT AND DISCUSSION

This section presents the results of the evaluation experiments that were conducted. Initially, we present all dependencies between labels discovered by a priori algorithm for each datasets. Then, the results of multi-label classifier methods on divided datasets are compared to traditional multi-label classifier methods without dividing the multi-label datasets, after that, comparing the results to ChiDep algorithm.

A. Dependencies between labels

Dependent labels were discovered by a priori algorithm and their respective support/ Actually, all generated labels combinations had their support $\geq .01$ for all datasets, except Cal500 dataset had support $\geq .05$.

In table II, medical dataset has 45 labels, only 7 labels (L5, L33, L45 , L25, L40, L37, L42) found to be dependent. The birds dataset identified 13 from 19 labels as dependent, where 20 label pairs have binary correlations such (L2 - L3) and only two label pairs have combination correlations such (L5 - L9 - L11). From this we observed that the birds dataset has correlations between some labels, and are (L2, L3, L8, L11, L12, L13, L9, L5, L7, L15, L14, L19, L10).

Similarly, for genbase datasets, 11 labels from 27 labels are dependent. The medical dataset has low dependencies between their labels. The situation was very similar for the scene dataset and Slashdot dataset. For yeast datasets, all 14 labels were found dependent. For Enron dataset, 30 labels from 53 are dependent. For cal500 datasets, 107 labels from 174 are dependent. Summarizing these results, we conclude that the degree of correlation depends on the support count.

B. The results after integration

This section presents the results of multi-label classifier algorithms after applying them on each data-subset and combining the results for all data-subsets. Integration is required and is done by the average function. Results are compared

Table II
NUMBER OF LABELS IDENTIFIED AS DEPENDENT FOR EACH DATASET FROM ASSOCIATION RULES.

	Labels	Dependent labels	Independent labels	Binary correlation pairs	Combination correlation
Scene	6	3	3	2	-
Yeast	14	14	-		
Birds	19	13	6	20	2
Slashdot	22	3	19	2	-
Genbase	27	11	16	10	3
Medical	45	7	38	4	-
Enron	53	30	23	85	102
Cal500	174	107	67	2000	12500

Datasets		LP	BR	CC	ECC	CLR	RAKEL	HOMER	PS	EPS
Scene	**Proposed**	0.0951	0.0973	0.0994	**0.0614**	0.1094	NF	0.0951	0.0942	0.0628
	Original	0.1437	0.1368	0.1444	0.0920	0.1383	0.1012	0.1418	0.1425	0.0974
Yeast	**Original**	0.2769	0.2454	0.2682	0.2041	0.2202	**0.2030**	0.2753	0.2799	0.2106
Birds	**Proposed**	0.0599	0.0460	0.0452	**0.0406**	0.0417	0.0437	0.0627	0.0576	0.0453
	Original	0.0735	0.0561	0.0562	0.0492	0.0506	0.0489	0.0788	0.0716	0.0508
Slashdot	**Proposed**	0.0678	0.0652	0.0690	0.0674	0.0737	NF	0.0676	0.0679	0.0584
	Original	0.0539	0.0423	0.0552	**0.0421**	0.0450	0.0490	.0529	0.0588	0.0443
Genbase	**Proposed**	0.0017	**0.0011**	**0.0011**	0.0014	0.0013	0.0012	0.0019	0.0017	0.0019
	Original	0.0019	**0.0011**	**0.0011**	0.0012	0.0013	0.0013	0.0021	0.0019	0.0020
Medical	**Proposed**	0.0136	0.0118	0.0118	0.0109	0.0129	0.0115	0.0135	0.0136	0.0120
	Original	0.0135	0.0103	0.0102	0.0098	0.0131	**0.0097**	0.0128	0.0127	0.0114
Enron	**Proposed**	0.0629	0.0480	0.0487	0.0436	0.0451	**0.0424**	0.0598	0.0598	0.0463
	Original	0.0708	0.0508	0.0524	0.0485	0.0471	0.045	0.0625	0.0635	0.0498
Cal500	**Proposed**	0.1693	0.1317	0.1347	0.1120	**0.1094**	0.1125	0.1538	0.1233	0.1188
	Original	0.1994	0.1615	0.1760	0.1437	0.1385	0.1357	0.1988	0.1487	0.1370

Figure 1. Evaluation of Hammimg Loss measure

Datasets		LP	BR	CC	ECC	CLR	RAKEL	HOMER	PS	EPS
Scene	**Proposed**	0.8390	0.7990	0.8265	0.8849	0.7918	NF	0.8390	0.8404	**0.8893**
	Original	0.5893	0.5353	0.5866	0.6702	0.5265	0.6247	0.5936	0.5924	0.6447
Yeast	**Original**	0.4144	0.4395	0.4280	**0.5221**	0.4685	0.5046	0.4032	0.4029	0.4958
Birds	**Proposed**	0.6708	0.7189	0.7223	**0.7349**	0.7316	0.7270	0.6731	0.6719	0.7026
	Original	0.4666	0.5295	.5222	0.5572	0.5280	0.5452	0.4600	0.4516	0.5139
Slashdot	**Proposed**	0.7204	0.6437	0.6820	0.6793	0.5015	NF	0.7182	0.7219	**0.7337**
	Original	0.4187	0.3869	0.3843	0.3620	0.4089	0.3854	.3426	0.4228	0.3722
Genbase	**Proposed**	0.9905	**0.9917**	**0.9917**	0.9888	0.9895	0.9914	0.9871	0.9905	0.9894
	Original	0.9826	0.9862	0.9862	0.9847	0.9857	0.9845	0.9794	0.9826	0.9804
Medical	**Proposed**	0.8573	0.8637	0.8624	0.8734	0.8278	**0.8737**	0.8560	0.8578	0.8709
	Original	0.7358	0.7465	0.7581	0.7773	0.6202	0.7774	0.7453	0.7476	0.7516
Enron	**Proposed**	0.4277	0.4851	0.4955	**0.5434**	0.5000	0.5317	0.4396	0.4456	0.5012
	Original	0.3460	0.4129	0.4233	0.4621	0.4209	0.428	0.3817	0.3667	0.4226
Cal500	**Proposed**	0.1813	0.2656	0.2714	**0.2978**	0.2863	0.2778	0.2280	0.2026	0.2745
	Original	0.2003	0.2067	0.2294	0.2377	0.1721	0.2500	0.2030	0.1956	0.2399

Figure 2. Evaluation of accuracy measure

with traditional multi-label classifier methods without having the dataset divided based on correlation between labels.

The performance of the methods is analyzed by the four defined evaluation measures using decision trees as a base classifier for all methods. The results are presented in figures 1-4. The NR mark indicates that there are no results for REKAL method because the number of labels divided by 2 is less than 3 and in RAKAL parameter k takes more or equal 3. For all measures, in yeast dataset all labels identified as dependent, so the results of original dataset are the same as datasets with proposed method applied upon. In figure 1 the hamming loss measure results are displayed, multi-label methods performed better on the proposed technique than the original datasets in 5 cases out of 7. In respect to the accuracy and subset accuracy measures, figure 2-3, the results of all multi-label methods on proposed datasets are more accurate than original datasets in all cases. In figure 4, the performance of all multi-label methods on proposed datasets are best in 4 cases for the micro-F measure. Moreover, we noticed that ECC algorithm performed better than the other multi-label classification techniques. Summarizing this comparison, we noticed that, as expected, all multi-label methods performed on most datasets with proposed technique better when evaluated by hamming loss, accuracy, subset accuracy and F-measure.

C. Evaluation of the algorithm against ChiDep algorithm

This section presents the comparison of results of proposed method to ChiDep algorithm. We used ECC method for the learning process on all datasets. The results are presented in tables III. According to the hamming loss measure the results of proposed method are better than ChiDep method in most cases. In respect to the accuracy measure, the results of proposed method are accurate than ChiDep method in all cases, as well as subsets accuracy and micro-F measures.

VI. CONCLUSION

In this paper, we proposed the methodology of treating multi-label datasets for classification for big data in a parallel framework. The basic idea is to divide the problem into subsets depending on the correlation between labels as classification is enhanced when learning is done separately. By using association rules for discovering dependencies between labels, information can be provided as the rules describe the correlations between labels. After that, each datasets correlated subset and uncorrelated subset are considered a problem solved separately in parallel. At the end all results are grouped to get the final classification results. The experimental results were evaluated using Accuracy, Hamming Loss, Micro-F-Measure and Subset Accuracy on a variety of datasets.

The experimental results revealed that dependencies between labels can be discovered by a priori algorithm, and the degree of correlation depends on support count. Through practical implementation we tested the proposed model, as expected the performance of all multi-label methods on proposed datasets are better for all measures, all correlations among labels in multi-label datasets exploit easily, facilitating the process of multi-label classification, increasing efficiency, accuracy, and performance.

Future work, The correlation between labels could further be investigated to find other exploiting techniques rather than association rules that might give better or faster results. The addition of confidence or fuzzy factors might provide better ranking of the rules. All these enhancement issues shall be further studied.

Table III
COMPARISON OF PROPOSED VS. CHI DEP ALGORITHM

Datasets	Hamming Loss		Accuracy		Subset Accuracy		Micro-avg F-measure	
	Proposed	ChiDep	Proposed	ChiDep	Proposed	ChiDep	Proposed	ChiDep
Scene	0.0614	0.1451	0.8849	0.5771	0.8645	0.5368	0.8278	0.604
Yeast	0.2041	0.2659	0.5221	0.4288	0.18	0.1494	0.6503	0.5692
Birds	0.0406	0.0934	0.7349	0.5103	0.6988	0.4974	0.1992	0.1539
Slashdot	0.0674	0.0436	0.6793	0.3873	0.658	0.3347	0.5035	0.4893
Genbase	0.0014	0.0009	0.9888	0.9916	0.9819	0.9849	0.9863	0.9897
Medical	0.0109	0.0117	0.8734	0.7252	0.8497	0.6547	0.8469	0.7936
Enron	0.0436	0.0538	0.5434	0.4114	0.3672	0.1295	0.5876	0.5194
Cal500	0.1120	0.1450	0.2978	0.1998	0.1276	0.0982	0.2934	0.2435
Average	*0.0677*	0.0949	*0.6906*	0.5289	*0.5910*	0.4232	*0.6119*	0.5453

Datasets		LP	BR	CC	ECC	CLR	RAKEL	HOMER	PS	EPS
Scene	**Proposed**	0.8241	0.7613	0.8077	0.8645	0.7404	NF	0.8241	0.8486	**0.8685**
	Original	0.5472	0.4266	0.5376	0.6273	0.3988	0.5604	0.5455	0.5521	0.6028
Yeast	**Original**	0.1357	0.0683	0.1531	**0.1800**	0.0972	0.1671	0.0778	0.1282	0.1605
Birds	**Proposed**	0.6320	0.6786	0.6849	0.6988	**0.6987**	0.6926	0.6317	0.6396	0.6863
	Original	0.3820	0.4469	0.4501	0.4841	0.4626	0.4749	0.3914	0.3850	0.4687
Slashdot	**Proposed**	0.6953	0.6242	0.6600	0.6580	0.4931	NF	0.6963	0.6977	**0.7086**
	Original	0.3689	0.3312	0.3264	0.3152	0.3968	0.3298	.2594	0.3675	0.3326
Genbase	**Proposed**	0.9849	**0.9856**	**0.9856**	0.9819	0.9834	0.9849	0.9804	0.9849	0.9827
	Original	0.9728	0.9713	0.9713	0.9683	0.9698	0.9698	0.9683	0.9728	0.9668
Medical	**Proposed**	0.8343	0.8363	0.8343	**0.8497**	0.8020	0.8497	0.8348	0.8354	0.8482
	Original	0.6635	0.6553	0.6778	0.6891	0.5315	0.6993	0.6716	0.6788	0.6788
Enron	**Proposed**	0.2632	0.3012	0.3194	**0.3672**	0.3152	0.3578	0.2553	0.2858	0.3328
	Original	0.1116	0.1028	0.1269	0.1445	0.1005	0.136	0.1010	0.1316	0.1386
Cal500	**Proposed**	0.0598	0.0976	0.1096	0.1276	0.1325	0.1079	0.0668	0.0733	**0.1350**
	Original	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Figure 3. Evaluation of subset accuracy measure

Datasets		LP	BR	CC	ECC	CLR	RAKEL	HOMER	PS	EPS
Scene	**Proposed**	0.7376	0.7272	0.7286	0.8278	0.7117	NF	0.7376	0.7408	**0.8253**
	Original	0.5982	0.6194	0.6001	0.7250	0.6276	0.6979	0.6060	0.6012	0.7042
Yeast	**Original**	0.5411	0.5857	0.5499	**0.6503**	0.6158	0.6369	0.5448	0.5317	0.6299
Birds	**Proposed**	0.1889	0.2000	0.1985	0.1992	0.1955	0.1995	0.1957	0.1794	0.1134
	Original	0.3258	**0.3747**	0.3514	0.3608	0.3127	0.3344	0.2584	0.2634	0.2495
Slashdot	**Proposed**	0.5350	0.4557	0.5046	0.5035	0.2004	NF	0.5290	0.5353	**0.5847**
	Original	0.4226	0.4964	0.4928	0.4753	0.5012	0.4963	.4424	0.4298	0.4592
Genbase	**Proposed**	0.9837	**0.9888**	**0.9888**	0.9863	0.9868	0.9883	0.9810	0.9837	0.9820
	Original	0.9801	0.9880	0.9880	0.9863	0.9862	0.9864	0.9782	0.9801	0.9789
Medical	**Proposed**	0.8162	0.8347	0.8343	**0.8469**	0.7772	0.8463	0.8134	0.8181	0.8382
	Original	0.7529	0.8091	0.8115	0.8226	0.7136	0.8187	0.7626	0.7643	0.7846
Enron	**Proposed**	0.4617	0.5429	0.5415	**0.5876**	0.5627	0.5788	0.4887	0.4731	0.5415
	Original	0.4364	0.5481	0.5363	0.5731	0.5672	0.548	0.4920	0.4528	0.5313
Cal500	**Proposed**	0.2119	0.3073	0.2935	0.2934	0.2702	0.2363	0.2818	0.2902	0.2857
	Original	0.3269	0.3396	0.3668	**0.3779**	0.2869	0.2789	0.3368	0.3245	0.3489

Figure 4. Evaluation of micro-avg F-measure

REFERENCES

[1] G. Tsoumakas and I. Katakis, "Multi label classification: An overview," *International Journal of Data Warehouse and Mining*, vol. 3, pp. 1–13, 2007.

[2] K. Dembczynski, W. Waegeman, W. Cheng, and E. Hüllermeier, "On label dependence in multi-label classification," in *Workshop proceedings of learning from multi-label data*. Citeseer, 2010, pp. 5–12.

[3] L. Chekina, D. Gutfreund, A. Kontorovich, L. Rokach, and B. Shapira, "Exploiting label dependencies for improved sample complexity," *Machine Learning*, vol. 91, no. 1, pp. 1–42, 2013.

[4] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," in *Proceedings of tECML PKDD '09: Part II*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 254–269.

[5] G. Tsoumakas and I. Vlahavas, "Random k-Labelsets: An ensemble method for multilabel classification," *Machine Learning: ECML 2007*, vol. 4701, pp. 406–417, 2007.

[6] J. Fürnkranz, "Round robin classification," *Journal of Machine Learning Research*, vol. 2, pp. 721–747, 2002.

[7] T. fan Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *Journal of Machine Learning Research*, vol. 5, pp. 975–1005, 2003.

[8] J. Read, B. Pfahringer, and G. Holmes, "Multi-label classification using ensembles of pruned sets," in *ICDM*. IEEE Computer Society, 2008, pp. 995–1000.

[9] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," in *Proceedings of tECML PKDD '09: Part II*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 254–269.

[10] G. Tsoumakas, I. Katakis, and I. P. Vlahavas, "Effective and efficient multilabel classification in domains with large number of labels," in *ECML/PKDD 2008 Workshop on Mining Multidimensional Data*, 2008, pp. 30–44.

[11] X. Kong, B. Cao, and P. S. Yu, "Multi-label classification by mining label and instance correlations from heterogeneous information networks," in *Proceedings of the 19th ACM SIGKDD KDD'13*. New York, NY, USA: ACM, 2013, pp. 614–622.

[12] J. Hipp, U. Güntzer, and G. Nakhaeizadeh, "Algorithms for association rule mining — a general survey and comparison," *SIGKDD Explor. Newsl.*, vol. 2, no. 1, pp. 58–64, 2000.

[13] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *In Data Mining and Knowledge Discovery Handbook*, 2010, pp. 667–685.

[14] G. Tsoumakas, E. Spyromitros-Xioufis, and J. Vilcek, "http://mulan.sourceforge.net/."

[15] U. of Waikato, "http://www.cs.waikato.ac.nz/ml/weka/."

Query-Based Dimensionality Reduction Applied To Images

Augustine S. Nsang, Donatus Edi and **C. Ahanonu**

[123]Department of Computer Science, School of Information Technology and Computing,
American University of Nigeria, Yola By-Pass, PMB 2250, Yola, Nigeria

Abstract

Real-world data such as digital photographs, fMRI scans and speech signals usually have high dimensionality. In other to handle these data adequately, their dimensionality needs to be reduced. Dimensionality Reduction is the transformation of high-dimensional data into a representation of this data of reduced dimensionality.

In this paper, we propose four new techniques to dimensionality reduction which shall include: the *New Bottom-Up Approach (*modified version), the *New Top-Down Approach* (modified version), and hybrids of these two novel approaches and the *Variance* Approach. We shall implement these four approaches as well as four existing approaches, which include *Random Projections (RP), Principal Component Analysis (PCA), LSA-Transform,* and *Variance*. We shall use these eight techniques to reduce images, and compare them with each other by the extent to which they preserve these images. Finally, we shall query the matrices representing the reduced images to find out certain features of the original images.

***Index Terms*— dimensionality reduction, image compression**

1. Introduction

Dimensionality Reduction is an interesting and important concept in Machine Learning. It enables us to reduce a data set containing *n* points in high dimensional space into a lower dimensional space, making sure the result obtained by working in the lower dimensional space is a good approximation to the result obtained by working in the original high dimensional space. In other words, given a *p*-dimensional data set $X = (x_1,\ldots,x_n)^T$, $x_i \in \Re^p$, we are to find a lower dimensional representation of it, $S = (s_1,\ldots,s_n)^T$, $s_i \in \Re^k$, with $k \leq p$, that captures the content in the original data, according to some criterion.

Dimensionality reduction has several advantages, the most important of which is the fact that with dimensionality reduction, we could drastically speed up the execution of an algorithm whose runtime depends exponentially on the dimensions of the working space (NR 2009). At the same time, the solution found by working in the low dimensional space is a good approximation to the solution in the original high dimensional space.

There are many known methods of dimensionality reduction. For some of these methods, each attribute in the reduced set is a linear combination of the attributes in the original data set. These include *Random Projection (RP), Singular Value Decomposition (SVD), Principal Component Analysis (PCA),* and many others (NR 2009)**.** Other dimensionality reduction methods, however, reduce a dataset to a subset of the original attribute set**.** These include the *Combined Approach (CA),* the *Direct Approach (DA),* the *Variance Approach (Var), LSA-Transform*, the *New Top-Down Approach (NTDn),* the *New Bottom-Up Approach (NBUp),* the *Weighted Attribute Frequency Approach(WAF)* and the *Best Clustering Performance Approach (BCP)* (NR 2010a).

One application of dimensionality reduction is to reduce the complexity of the query process. Suppose, for instance, that we want to query a text document database with say 5000 dimensions. It would be helpful if we can reduce it to 400 dimensions, say, before applying the query, provided the dimensions in the query are not eliminated by the dimensionality reduction process. The complexity of the query processing is reduced while the speed is significantly increased (NR 2011).

Another application of dimensionality reduction is in compression of image data. In this domain, digital images are stored in 2D matrices which represent the brightness of each pixel. Usually, the matrix representing an image can be quite large, and for this reason it could be very time consuming querying this matrix to find out any information about the features of the reduced image (NBH 2015).

In this paper, we shall implement eight dimensionality reduction techniques and use them to reduce images. We shall then query the reduced images to find out certain features of the original images.

The rest of this paper is organized as follows. In Section 2, we shall describe the four novel approaches we have proposed. In Section 3, we shall implement these four approaches as well as the four existing approaches listed above. We shall use them to reduce images and compare them with each other by the extent to which they preserve these images. In Section 4, we shall query the matrices representing the reduced images to find out certain features of the original images. Then we shall conclude this paper in Section 5.

2. Our Novel Techniques

In this section, we propose four new techniques that can be used to reduce the dimensionality of a dataset. They include the following:

2.1 The New Top-Down Approach (Modified Version)

The New Top-Down Approach was proposed by Nsang and Ralescu in (NR 2010b). It considers subsets of attributes reduced by one attribute at a time. To reduce a data set *D*

containing *p* attributes to a data set D_r containing *r* attributes, the Top-Down Approach starts with the original *p* attributes of D, and reduces it successively to the subset of *p – i, i = 1, 2, ...* attributes which best preserve k -means clustering.

In this paper, we propose a modified version of this approach which successively reduces a set of *p* attributes to the subset of *p – i, i = 1, 2, ...* attributes which best preserve the interpoint distances (rather than the *k*-means clustering) of the original dataset. Algorithm 1 shows the modified version of the New Top-Down Approach to reduce the original set of *p* attributes to a subset of *r*.

Algorithm 1: The New Top-Down Approach (Modified Version)

Input: *D* (a data set with *n* rows and *p* columns)

Output: *D1* (a data set with *n* rows and *r* columns, $r < p$)

1. Generate *LC*, the list of the *p* combinations of *p - 1* attributes of *D*.
2. Assign to *t* the value of *p – 1*.
3. While $t \geq r$:
   (a) Compute the extent to which each combination of attributes in *LC* preserves the interpoint distances of *D*. Store the result in a new list, *AML*
   (b) Find the maximum element of *AML*, and thus the combination L_0, of *t* attributes of *D* which best preserves the interpoint distances
   (c) Reset *LC* to be the list of the *t* combinations of *t – 1* attributes of L_0
   (d) Reset $t \leftarrow t - 1$

   Endwhile;
4. The result, *D1*, is given by: $D1 = D(:, L_0)$

Note: Suppose a dataset *D* is reduced to a dataset *D1* by a given dimensionality reduction method *M*. Then the extent to which the interpoint distances of *D* are preserved by *M* is given by:

$$\sum_{u,v \in rows(D)} \frac{\| f(u) - f(v) \|^2}{\| u - v \|^2}$$

where *u* and *v* are any two rows of *D*, and *f(u)* and *f(v)* are the corresponding rows in *D1*.

2.2 The New Bottom-Up Approach (Modified Version)

The New Bottom-Up Approach was also proposed by Nsang and Ralescu in (NR 2010b). Contrary to the *New Top-Down* dimensionality reduction approach, the *New Bottom-Up* approach considers subsets of attributes increased by one attribute at a time. Starting with the subset S_1 containing the single attribute, i_1, of D, which best preserves k -means clustering, it increases it to the subset S_2 of two attributes (i_1, i_2) which best preserve *k-means* clustering. The process continues until the subset S_r of r attributes $(i_1, \ldots, i_r)$ which best preserve k -means clustering has been obtained.

Again, in this paper, we propose a modified version of this approach which selects the *r* attributes which best preserve the interpoint distances rather than the *k-means* clustering of the original dataset.

Algorithm 2 shows the modified version of the New Bottom-Up Approach to reduce the original set of *p* attributes to a subset of *r*.

Algorithm 2: The New Bottom-Up Approach (Modified Version)

Input: *D* (a data set with *n* rows and *p* columns)

Output: *D1* (a data set with *n* rows and *r* columns, $r < p$)

1. Compute the vector *AML* such that *AML(i)* represents the extent to which the i^{th} attribute of *D* preserves the interpoint distances of *D*
2. Find the maximum element of *AML*, and thus find *x*, the attribute of *D* which best preserves the interpoint distances of *D*
3. Take $L_0 = [x]$, and $t = 1$.
4. While $t < m$:
   (a) Generate all $p - t$ combinations of $t + 1$ attributes of *D* which include the *t* attributes in L_0. Let *LC* be the list containing these $p - t$ combinations
   (b) Redefine *AML* to be the list containing $p - t$ values such that *AML(i)* represents the extent to which the i^{th} combination of attributes in *LC* preserves the interpoint distances of *D*
   (c) Find the maximum value of *AML*, and thus the combination, *L*, of attributes of *D* (in *LC*) that best preserves the interpoint distances of *D*
   (d) Increase *t* by 1, and redefine L_0 to be the list of attributes in *L*

   Endwhile;
5. The result, *D1*, is given by: $D1 = D(:, L_0)$

2.3 The Variance – New Top-Down Hybrid

As you would expect, the New Top-Down Approach (and its modified version) are very efficient in preserving the interpoint distances of the original dataset and related properties such as *k-means* clustering and *k-nearest neighbors* classification. However, it is very very slow to run. Because of its very low speed, it is an undesirable approach for compressing images which are usually represented by very large matrices. For this reason, we propose a hybrid of this approach and the *Variance* approach. To reduce a dataset containing 200 columns to 60 columns, say, the Variance–New Top-Down Hybrid first uses the *Variance* approach to reduce the dataset from 200 columns to say 120 columns, and would then use the modified version of the New Top-Down Approach to reduce the dataset from 120 columns to 60 columns. Obviously, this hybrid will, in general, be more efficient in preserving the interpoint distances and related properties of the original dataset than the *Variance* approach, and be much faster than the modified version of the New Top-Down Approach.

2.4 The Variance–New Bottom-Up Hybrid

For the same reason as in Section 2.3, we propose a hybrid of the (modified version of) the *New Bottom-Up* approach and the *Variance* approach. Again, this hybrid will, in general, be more efficient in preserving the interpoint distances and related properties of the original dataset than the *Variance* approach, and be much faster than the modified version of the *New Bottom-Up* Approach.

3. Image Reduction Using the Dimensionality Reduction Techniques

The eight techniques mentioned above: *RP, PCA, Variance, LSA-Transform, New Top-Down Approach (modified version), New Bottom-Up Approach (modified version),* the *Variance-New Top-Down* hybrid and the *Variance-New Bottom-Up* hybrid were used to reduce an image. To achieve this, the MATLAB function *imread* was first used to obtain the matrix representing the image. The dimensionality reduction technique was then applied to reduce this matrix, after which the MATLAB function *imshow* was used to display the compressed image (i.e. the image represented by the reduced matrix). The results of reducing this image using these techniques are displayed below:

3.1 RP and PCA:

As was realised by Nsang, Musa and Shamsuddeen in their paper (NBH 2015), we found out that these two approaches are useless in preserving images. This applies to most approaches in which each attribute in the reduced set is actually a linear combination of the attributes in the original dataset.

3.2 The Other Methods

All the other six techniques were efficient in reducing this image. The results obtained by each approach are outlined here.

3.2.1 LSA-Transform

Original Image:

Reduced Image:

3.2.2 Variance

Original Image:

Reduced Image:

3.2.3 The New Top-Down Approach (Modified Version)

Original Image:

Reduced Image:

3.2.4 The New Bottom-Up Approach (Modified Version)

Original Image:

Reduced Image:

3.2.5 The Variance-New Top-Down Hybrid

Original Image:

Reduced Image:

3.2.6 The Variance-New Bottom-Up Hybrid

Original Image:

Reduced Image:

Our experiments show that:

- Of the six approaches, *LSA-Transform* is probably the best in preserving images
- All the other five approaches are reasonably efficient in preserving images. However, when the number of attributes in the matrix representing the reduced image is significantly smaller than the number of attributes representing the original image, some of these techniques perform clearly better than others. Of the five techniques, the (modified versions of) the *New Top-Down* and *New-Bottom Up* are more efficient than the other three (in fact almost as good as *LSA-Transform)*, and the two hybrids are not as efficient as the modified versions of the *New Top-Down*

and *Bottom-Up* Approaches but more efficient than the *Variance* Approach.

- *LSA-Transform* also has the highest speed of execution, followed by the *Variance* Approach. Of the six approaches, these are the only two which are suitable for reducing large images. The two hybrids are much faster than the modified versions of the *Top-Down* and *Bottom-Up* Approaches, but very slow compared to *LSA-Transform* and the *Variance* Approach.

4. Querying the Reduced Images (Nsang 2011)

Given an image, *I*, we have seen in the previous sections how to:

- find a presentation of *I* as a 2D-matrix, *M*, of pixel brightness values
- compress *M* to a matrix *M1* much smaller than *M* which which represents the same image *I*.

Our goal in the section is to see how we can query the reduced matrix *M1* to find out certain features of *I*. Sample queries include:

- Compare the brightness of the top half of this image (on the average) to that of the bottom half
- Compare the brightness of the left half of this image (on the average) to that of the right half
- Verify that there is a thick horizontal line through the middle of this image.
- Verify that the image has a thick right border
- Verify that there is a blank portion at the top left of this image.

In this paper, we used each of the six techniques discussed above to implement these queries, and in each case we obtained positive results. The algorithms we implemented for these queries are outlined below:

Query 1: *Compare the brightness of the top half of this image (on average) to that of the bottom half*

Algorithm:

1. Find the sum of the entries in the top half of the matrix *M1*. Call it *sum1*.
2. Find the sum of the entries in the bottom half of the matrix *M1*. Call it *sum2*.
3. If *sum1 > sum2*,
   declare the top half is brighter, on the average
   Else if *sum1 = sum2*
   declare the two halves are equally bright (on the average)
   Else
   declare the bottom half is brighter, on the average

Query 2: *Compare the brightness of the left half of this image (on the average) to that of the right half*

Algorithm:

1. Find the sum of the entries in the left half of the matrix *M1*. Call it *sum1*.
2. Find the sum of the entries in the right half of the matrix *M1*. Call it *sum2*.
3. If *sum1 > sum2*,
   declare the left half is brighter, on the average
   Else if *sum1 = sum2*
   declare the two halves are equally bright (on the average)
   Else
   declare the right half is brighter, on the average

Query 3: *Verify that there is a thick horizontal line through the middle of this image*

Algorithm:

1. Start at the point *M1(r1, 1)* (where *r1 = round(r/2)*)
2. Set *thl = 1*
3. Keep moving to the next cell on the right until you meet a cell *A(r1, x)* whose pixel brightness value ≤ 95
4. If *(x – 1)/c > 0.15* Then
   set *thl = 0*
5. If *thl = 1* then
   While the current cell and the two cells above and below it have pixel brightness values < 95
   Move to the next cell on the right
   Move back one cell to the left to cell *B(r1, y)*
6. If (c – y)/c > 0.15 then
   thl = 0
7. If thl = 1
   declare that there is a thick horizontal line through the middle of this image
   Else
   declare that there is no thick horizontal line through the middle of this image

Query 4: *Verify that the image has a thick right border*

Algorithm:

Assuming the matrix *M1* has *r* rows and *c* columns:

1. Start at the point *M1(r1, c)* where *r1 = round(r/2)*

2. Keep moving to the next cell on the left until you find the first cell, *C(r1, x)*, with pixel brightness value less than 95
3. From *C*, keep moving to the next cell above until you find the first cell, *D(j, x)*, with pixel brightness value ≥ 95. Then move back one cell below to *E(y, x), y = j + 1*
4. From *C*, keep moving to the next cell below until you find the first cell, *F(q, x)*, with pixel brightness value ≥ 95. Then move back one cell above to *G(k, x), k = q – 1*
5. If ((j/r) < 0.15) & ((r – k)/r < 0.15) & ((c – x)/c <0.15) Then
   If each cell in columns *x – 1, x – 2* and *x – 3* (rows *y* to *k*) has a pixel value < 95 Then
      Declare the image has a thick right border
   Else
      Declare the image does not have a thick right border

Query 5: *Verify that there is a blank portion at the top left of this image*

Algorithm:

1. Start at the point M1(1,1)
2. Check each cell in the first one-eighth of the rows and in the first one-eighth of the columns (i.e. from rows 1 to *r1* and columns 1 to *c1*, where *r1 = round(r/8)* and *c1 = round(c/8).*) If any of them has pixel brightness value less than 150, then
   declare that there is no blank portion at the top left of this mage
   else
   declare that there is a blank portion at the top left of this mage

Note:

Other sample queries include:

- Does the image have a bottom border with thick dotted lines?
- Is there a thick dotted vertical line near the right end of the image?
- Compare the brightness of the top quarter of this image (on the average) to that of the bottom quarter
- Verify that thery is a very dark letter 'H' somewhere in this image
- Verify that there is a thick dotted vertical line through the middle of this image
- Verify that there ia thick border al round the image

The MATLAB implementation of all these eleven queries can be found in (Nsang 2011).

5. Conclusion

In this paper, we have proposed four novel techniques to dimensionality reduction, which include modified versions of the *Top-Down* and *Bottom-Up* Approaches (see (NR 2010b)), and hybrids of these two approaches with the *Variance* Approach. We have implemented these approaches alongside four others: *RP, PCA, LSA-Transform* and *Variance* and used them to reduce images.

As was observed by Nsang, Bello and Shamsudeen in (NBH 2015), we found that:

- *RP, PCA* and most other approaches in which each attribute in the reduced set is a linear combination of the attributes in the original dataset is useless in preserving images
- All other six approaches are reasonably efficient in preserving images, but *LSA-Transform* is probably the best technique taking into consideration both efficiency and run-time complexity.

Finally, we examined several queries that can be applied on the reduced image matrices to find out certain features of the original image. We found that for each of the six techniques mentioned above, each query on the reduced matrix correctly returned the features of the original matrix. For instance, if the original image had a thick border all round it, we could deduce that by querying the matrix obtained by reducing the matrix representing the original image using any of the six dimensionality reduction techniques.

References

Nsang A., Ralescu A. 2009, A Review of Dimensionality Reduction and Their Applications. In *Proceedings of the Twentieth Midwest Artificial Intelligence and Cognitive Science Conference (MAICS 2009), Fort Wayne, IN.,* 118 - 123.

Nsang A., Ralescu A. 2010, Approaches to Dimensionality Reduction to a Subset of the Original Dimensions. In *Proceedings of the Twenty-First Midwest Artificial Intelligence and Cognitive Science Conference (MAICS 2010), South Bend, IN.,* 70 - 77.

Nsang A., Ralescu A. 2011, A Study on Query-Based Dimensionality Reduction. In *Proceedings of the Twenty-Second Midwest Artificial Intelligence and Cognitive Science Conference (MAICS 2011), South Bend, IN.,* 70 - 77.

Nsang A., Bello A., Shamssuddeen H. 2015. Image Reduction Using Dimensionality Reduction Techniques. In *Proceedings of the Twenty-Sixth Modern Artificial Intelligence and Cognitive Science Conference (MAICS 2015), South Bend, IN.*

Nsang A., Ralescu A. 2010. More Dimensionality Reduction to a Subset of the Original Attribute Set. In *Proceedings of the Twenty-First Midwest Artificial Intelligence and Cognitive Science Conference (MAICS 2010), South Bend, IN.,* 109-116.

Nsang, A. 2011. *Novel Approaches to Dimensionality Reduction and Applications, An Empirical Study*. Lambert Academic Publishing.

SESSION

APPLICATIONS OF BIG DATA + SOFTWARE TOOLS

Chair(s)

TBA

Who Influences Whom: Content-based Approach for Predicting Influential Users in Social Networks

Khaled Almgren, and Jeongkyu Lee
Department of Computer Science and Engineering, University of Bridgeport, Bridgeport, CT 06604
kalmgren@my.bridgeport.edu, jelee@bridgeport.edu

Abstract— *Identifying influential users is useful in many real-world applications including viral marketing, recommendation systems and expert search engines. In this paper, we propose a Content-Based Influence Measurement, i.e., CIM, to predict influential users in social networks that considers both content and context. We apply Context-Based Influence Measurements that consider static social network structure, such as centrality analysis, to evaluate the robustness of our approach. We define influence as the total amount of social interactions a user receives through his/her posts, which does not only reflect the user popularity but the interaction strength between influential users and his/her followers, as well as the quality of the influential users posts. The results on 7394 Flickr users show the superiority of the proposed approach over the other approaches. It also shows that followers have different interaction strength with influential users, and content alone does not predict influential users.*

1 Introduction

In 1962, Everett Rogers said that finding influential users is the best way to persuade others [6]. Social influence was first studied a long time ago by Leo Katz [11]. According to [23], influence is the power to generate an effect on people without using any force. This effect can be feelings, thought or actions [14]. It is very challenging to measure influence in real world. However, social networks have made it easier to study influence.

Recently, the amount of research on social network analysis has increased tremendously because of the dramatic increase in the number of social network users. Social networks are very useful tools in information spread and social activities [6]. Therefore, people have been using social networks very frequently to communicate with each other, to spread ideas, or even to discuss different topics. The huge increase in the social network usage has attracted many researchers to study many social phenomena such as influence. One of the major issues in research is to identify influential users. Identifying influential users can bring value to many important real-world applications, such as viral marketing, recommendation systems, and expert search engines. [3], [5], [8], [12]–[15], [18]–[21]

In viral marketing, companies promote their products using the word-of-mouth model. Social network sites such as Twitter, Facebook, and Flickr have a massive user base of hundreds of millions of connected users who use their sites on a daily basis. Therefore, companies can use these social networks to reach out to millions of users. However, it is not feasible to contact all users, it is more practical to use the influential users, who already have an impact on many users, to target consumers. But the struggle is selecting the most influential user.

Social networks such as Facebook, Twitter, and Flickr are based on content and context, where context is the social network structure. Therefore, measuring influence should consider both. Influence can be measured from the users actions toward each other. These actions are called social interactions, such as *retweet* on Twitter, *favorite* on Flickr, or *like* on Facebook, where all these social interactions can depend on the content which can present a significant indicator of influence. These social interaction that show influence can be presented as ties in social network. In network theory, ties can be considered as edges structure. For example, ties can be explained as *who-likes-whom* or *who-follow-whom* [14]. These ties have different strengths that can indicate different amount of influence. This can be considered as the tie strength between different users. We define influential users in social network as users who can influence others to perform further social interactions. The amount of social interactions are quantified as INR. INR is the number of social interactions between two users. The social interactions in social networks can be a *comment*, *favorite*, *retweet*, or *like*.
In this paper, we consider three factors in measuring influence:

(1) The content,
(2) How well the user is connected in social networks, and
(3) The influence strength between users.

For example, as shown in Figure 1, Bob and Alice both have a follower Charlie. Bob, Alice, Charlie $\in V$, where V is a set of users. Let Charlie *like* the posts of Bob two times $(INR = 2)$, and Alice one time $(INR = 1)$. Let INR represent influence where $Influence = INR$ therefore, $Influence(Bob) > Influence(Alice)$. Our problem is formalized as below:
Social Influence: For any social network, called SN that is represented as a weighted directed graph $G(V, E)$ of V vertices and directed weighted E edges, given a set of connected users $V_n = \{v_1, \ldots, v_n\}$ who can post multiple posts $p \in P$, where P is a set of posts, and perform multiple social interactions $z \in Z$, where Z is a set of social interaction shown as edges $E_n = \{e_1, \ldots, e_n\}$ that can have different strength quantified as INR calculated as $w(e_{i,j})$, predict a user i who will influence more users j to perform further social interaction Z.

Unlike previous research which is based on Benchmark data or ground truth, this research does not have ground truth which presents a significant challenge for researchers that made it difficult to evaluate their approaches. To address these problems, several studies have applied their influence measures to real-world social networks [3].

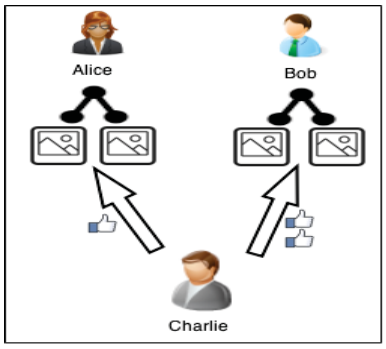


Fig. 1
SOCIAL INFLUENCE FLOW

The main contributions are listed below:

(1) We propose a Content-Based Influence Measurement, i.e., CIM, that considers both content and context properties.
(2) We prove the robustness of CIM by applying CIM and Context-Based Influence Measurements to Flickr.
(3) We evaluate CIM using a predictive approach to predict the accuracy of CIM by considering the similarity between users based on Cosine similarity. This approach can also be used to predict social interactions.

The remainder of the paper is organized as follows. We review the related work and categorize them according to the influence model in section 2. In section 3, we present our influence model. We propose CIM and discuss the current Context-based Influence Measurements in section 4. In section 5, we discuss our experimental setup.

2 Related Work

Identifying influential users in SN can be measured using prediction-based measures or observation-based measures. In prediction-based measures, researchers use network measures, such as centrality analysis, to predict influential users in SN. In observation-based measures, researchers measure the amount of influence attributed to users. [22]

2.1 Prediction-based measures

Ghosh and Lerman define influence as the average number of votes a story receives on Digg [9]. Being one of the first researchers to use centrality analysis to predict influential users, their findings show that considering the depth is the best measure of influence [9]. Cha et al. measured influence by correlating the number of followers, number of retweets, and the number of mentions in different trending topics and time on Twitter [5]. Weng et al. proposed a modified algorithm based on PageRank to find the influential users on specific topics. They consider Homophily in finding influential users [21]. Li et al. try to find influential users on Twitter based on the dynamic information diffusion model. They consider the temporal order of influential users' tweets [14]. Leavitt et al. were the first to analyze influence in SN. They track 12 users on Twitter over a period of ten days [16]. [2]define influential users as users who have a large active audience that consume and multiply the published content posted by them. They ranked influential users based on the ratio of (followers/following) and (retweet/mention).

2.2 Observation-based measures

Lee et al. consider the temporal adoption of information on Twitter as a measure of influence [13]. Sun and Vincent identify influential users as the ones who start a topic by sharing a tweet that attracts many users to explicitly or implicitly make interactions [20]. Bakshy et al. look into the problem of finding influential users from the angle of information diffusion. They identify an influential users as a user who shares a URL on Twitter that attracts many re-shares from her direct and indirect followers [3].

As far as we know, most of the current research have used Context-Based Influence Measurements where we propose a Content-Based Influence Measurement that considers both content and context. In addition, most of the papers assume that the influence of influential users on her followers are of the same strength where in SN, influence between influential users and her followers can have different strengths. Moreover, no one has tried to predict social interactions to evaluate their approach in term of accuracy or prove the robustness of the used approach.

3 Influence Model

We use graph theory and two matrices to represent the influence model. First, SN is a network of nodes and edges that are represented as vertices V and directed-weighted edges E, respectively. Users are represented as V, they can be contributed users, follower or both. User v_i is categorized under contributed users if v_i shares posts P in SN. User v_j is a follower of v_i, if v_j interact with v_i by performing social interactions on v_i posts P. V can be weighted to exhibit the number of P. For example, if Bob shares two posts, $w(v_{Bob}) = 2$. The social interaction is represented by a directed-weighted edge $e_{i,j}$ between i and j. They can be comments $c \in C$, where C is a set of comments, and favorites $f \in F$, where F is a set of favorites. To avoid E redundancy, all social interactions are represented by a single e. $E_n = \{e_1, \ldots, e_{|n|}\}$ are quantified in term of INR between two users. To compute $w(e_{i,j})$, we count the number of social interactions that v_i performed on v_j posts. $w(e_{i,j})$ also shows the interaction strength, for example if Charlie favorites Bob P five times $INR(Bob, Charlie) = w(e_{Charlie,Bob}) = 5$. See Table 1 for the notations.

To represent the influence model, we use two matrices. The Weighted Adjacency Matrix, called M and Degree Matrix, called D. M is $|V| \times |V|$ matrix that represent the number

of social interactions between the users in SN. It computes $w(e_{i,j})$ between each i and j as discussed above. Let $M = M[i,j]$ be the weighted adjacency matrix of SN. $M[i,j] = W(e_{i,j})$, where $\exists\ e_{i,j}\ \forall\ i,j \in V$, as shown in Equation 1.

$$Mij := \begin{cases} w(e_{i,j}) & if\ there\ is\ edge\ from\ i \rightarrow\ j \\ 0 & if\ there\ is\ no\ edge\ from\ i \rightarrow\ j \end{cases} \quad (1)$$

Algorithm 1 depicts the process of creating M. In line 1,we initialize $M[i][j]$ with size $|V+1| \times |V+1|$ to represent the users identifiers, called IDs, in the first row and column. Between lines 2 and 5, we label $M[0][i] \leftarrow v.id$. Between lines 6 and 14, we compute the weight of $e_{i,j}$ and add it to $M_{i,j}$, if $e_{i,j}$ exists with weight w that satisfy the condition in line 9.

D is $V \times V$ a diagonal matrix that represent the number of followers based on the $deg(v_i)$. We compute $deg(v_i)$ by counting the number of followers each user has. Let D be the diagonal degree matrix where $D[i,j]$=$deg(v_i)$ if $i = j$ as shown in Equation 2, where d_i^{in} shows the in-degree of i, and d_i^{out} shows the out-degree of i. To create D, we use a similar algorithm is that of Algorithm 1, but we store $D_{i,j}$ in the diagonal $D[i,i]\ \forall\ v_i$ we count the adjunct vertices $v_j\ \exists\ e_{i,j}$.

$$Dij := \begin{cases} deg(v_i) & if\ i = j \\ 0 & if\ i \neq j \end{cases} \quad (2)$$

Algorithm 1: Create Weighted Adjacency Matrix

```
   Data: V: Nodes list E: Edges list
   Result: M: Weighted Adjacency Matrix
1  M[v+1][v+1] ← 0;
2  for all i ∑ V do
3  |  let M[0][i] ← 0;
4  |  let M[i][0] ← 0
5  end
6  for all v_i ∑ M+1 do
7  |  for all v_j ∑ M+1 do
8  |  |  for all edges x ∑ E do
9  |  |  |  if v_i = x_i.SourceNode ∩
   |  |  |  v_j=x_i.DestinationNode then
10 |  |  |  |  M[i][j]← x_i.weight
11 |  |  |  end
12 |  |  end
13 |  end
14 end
15 Return M;
```

4 Influence Measures

In this section, we discuss the Context-Based Influence Measurements and then we extend to the Content-Based Influence Measurement which is based on the previous models. Because these measurements are based on centrality analysis, we first present the Linton Freeman theorem in centrality analysis [7].

Theorem 1: The centrality of the entire network is defined by two factors no matter what centrality measure it is based on. (1) It should indicate the most central node where it has the highest centrality over all nodes. (2) The measure should be represented as the average difference between the most central node and the other nodes. [7]

Table 1
NOTATIONS

Notation	Meaning
SN	Social Network
$G(V,E)$	Graph
$V_n = \{v_1, v_2, , , , v_n\}$	Vertices
$E_n = \{e_1, e_2, , , , e_n\}$	Edges
$Z_n = \{z_1, z_2, , , , z_n\}$	Social interaction
$P_n = \{p_1, p_2, , , , p_n\}$	Posts
$F_n = \{f_1, f_2, , , , f_n\}$	Favorites
$C_n = \{c_1, c_2, , , , c_n\}$	Comments
$w(e_{i,j})$	Weight of edge
$w(n_i)$	Weight of node
INR	Strength of interaction
$Sim(x,y)$	Cosine similarity function
$SLoc(x,y)$	Location similarity function
$TSim(u1,u2)$	Total similarity function between user 1 and user 2

$$C_X = \frac{\Sigma_{i=0}^{n}[C_X(P^*) - C_X(P_i)]}{Max\Sigma_{i=0}^{n}[C_X(P^*) - C_X(P_i)]} \quad (3)$$

,where $C_X(P^*)$ is the maximum centrality in the network, $C_X(P_i)$ is the centrality measure of any point in the network, and Max $\Sigma_{i=0}^{n}[C_X(P^*) - C_X(P_i)]$ is the maximum possible sum of differences between $C_X(P^*)$ and $C_X(P_i)$. A graph is perfectly centralized if $C_X = 1$ where $0 \leq C_X \geq 1$. Freeman showed and proved that centrality of a graph can be calculated using Equation 3 for several centrality analysis measures. [7]

Proof. Omitted. See [7] for the proof.

4.1 Context-Based Influence Measurements

Many graph mining algorithms can be used to analyze interaction patterns, and finding important nodes. Centrality analysis is used to find central nodes V that can present such importance over other nodes based on SN structure. There are various kinds of Context-Based Influence Measurements including Degree Centrality, Betweenness Centrality, and PageRank Centrality [23]. See Table 2 for the notations.

Table 2
CENTRALITY ANALYSIS ALGORITHMS

Notation	Centrality Function
C_X	Graph Centrality
$C_{\mathbf{d}}$	**P**ageRank Centrality
$C_{\mathbf{b}}$	**B**etweenness Centrality
$C_{\mathbf{p}}$	**D**egree Centrality
CIM	**C**ontent based Influence Measurement, the proposed measurement

4.1.1 Degree Centrality

In real-world, we always think of important people as people with the most number of friends. $C_{\mathbf{d}}$ uses this idea

in ranking V. For example, if number of friends of $i > j$, $INR(i) > INR(j) \therefore i$ is more important than j. There are two types of $C_{\mathbf{d}}$, in-degree, i.e., d_i^{in} and out-degree, i.e., d_i^{out}. d_i^{in} shows how popular a node is by considering only the incoming E. for example, Bob and Alice can have different number of followers, while d_i^{out} shows the nearness of nodes by considering the outgoing E, for example Charlie follow Bob and Alice. $C_{\mathbf{d}}(v_i) = d_i^{in} + d_i^{out}$. We can also consider computing only d_i^{in} or d_i^{out}, where d_i^{in} of i show the number of follower and d_i^{out} show the number of *followings* such as on Twitter. To get d_i^{in} and d_i^{out}, we compute D from equation (2). However, in this measure, we only consider d_i^{in}, where $d_i^{out} = 0 \therefore C_{\mathbf{d}} = d_i^{in}$. [23]

4.1.2 Betweenness Centrality

Intermediate people are important in the real world. They connect other people together. $C_{\mathbf{b}}(v_i)$ rank V in term of how many nodes s and t they connect. Let $\forall\ j$ that comment or favorite on s that go through i, where i connects j and s. We compute the number of shortest paths between s and j that go through node i [23]. For example, if Charlie has two followers, he would have the highest centrality since he is the only one who can connect all the users. Let $C_{\mathbf{b}}(v_i) = \Sigma_{j \neq s \neq i} \frac{\sigma_{st}(v_i)}{\sigma_{st}}$, where $\sigma_{st}(v_i)$ is the shortest paths between any j and s that pass through i, and σ_{st} is the number of shortest paths between j and s. We normalize $C_{\mathbf{b}}$ by dividing it by max $C_{\mathbf{b}}(v_i)$ as shown in Equation 4 according to [23]. To compute this algorithm, we implement the algorithm proposed in [5].

$$C_{\mathbf{b}} = \frac{C_{\mathbf{b}}(v_i)}{max(C_{\mathbf{b}}(v_i))} \tag{4}$$

4.1.3 PageRank Centrality

In contrast to the degree and Betweenness centrality measures, $C_{\mathbf{p}}(v_i)$ considers the important connections. According to $C_{\mathbf{p}}(v_i)$, v_i who has important followers is more important than v_j, who has unimportant followers. For example, Bob would have higher centrality than Charlie if he has another follower Sally that has at least one follower. However, ranking nodes based on $C_{\mathbf{p}}(v_i)$ is more complicated than the previous measures. As shown in Equation 5, we compute $C_{\mathbf{p}}(v_i)$ by $v_i \times A \times \Sigma C_{\mathbf{p}}(v_i)$ divided by outgoing edges d_i^{out} to mitigate the effect when V with a high $C_{\mathbf{p}}(v_i)$ passes its $C_{\mathbf{p}}(v_i)$ to the adjacent j through its outgoing E, where α is a constant that is used to avoids zero centrality values while β is a constant used as an attenuation factor. A is the adjacency matrix. We implement the PageRank algorithm based on the popular Google PageRank used in [23].
Let $C_{\mathbf{p}}(v_i) = \alpha \sum_{i=0}^{n} A_{ij} \frac{C_{\mathbf{p}}(v_i)}{d_i^{out}} + \beta, (d_i^{out} > 0)^3, \beta > 0$
$C_{\mathbf{p}}(v_i)$can be rewritten in a matrix format
$C_{\mathbf{p}}(v_i) = \alpha \times A \times D^{-1} \times C_{\mathbf{p}}(v_i) + \beta I$

$$C_{\mathbf{p}}(v_i) = \beta(I - \times AD^{-1}C_{\mathbf{p}}(v_i)^{-1} \times 1) \tag{5}$$

4.2 Content-Based Influence Measure

Although Context-Based Influence Measurements are popular and have been used widely to find influential users, it does not consider the power of users in their measures. Therefore, to address this limitation, we propose a Content-Based Influence Measurement that is based on both content and context. Since CIM considers the SN structure, it can be represented based on the Linton theorem.

Corollary 1:Freeman's theorem can be extended to include directed and weighted graph. The max possible centrality for any node in the graph based on CIM would be $CIM(v_i) = Max\left(d_{j=0}^{w(v-1)} v_i\right)$, where the minimum possible $CIM(v_i) = 0$.

$$CIM = \frac{\Sigma_{j=0}^{V-1}[CIM(P^*) - CIM(P_i)]}{Max\left(d_{j=0}^{w(V-1)} v_i\right) \times V - 1} \tag{6}$$

Proof. Omitted because of space limitation.
In SN, when v_i shares $P_n \{= p1, \ldots, p\}$, other users $\{v_1, \ldots, v_n\}$ in the SN can interact with v_i by making a social interaction on v_i posts. Note that v_i can share many posts. For example, let us assume that a group exists on Flickr, which has three users Bob, Alice, and Charlie. Assume Bob shared p. Now Alice and Charlie can view p_1, and then they can either favorite f_1 leave a comment c_1 on p_1 or do nothing. If they perform a social interaction, there will be $e_{Charlie,Bob}$, $e_{Alice,Bob}$, where $w(e_{Alice,Bob})$ present the INR strength between Bob and Alice. These social interaction can be based on the influence of Bob. Social interactions such as Favorites $\{f_1, \ldots, f_n\}$ can imply that Alice and Charlie *like* the content that is shared by Bob, and Comments $\{c_1, , c_n\}$ can also imply the same thing. When Bob shares multiple P that are *liked* by many users, this user can be categorized as an influential users. The posts that receive many *likes* can be considered of a high quality. Therefore, INR depends on both content and context. To measure the influence, let CIM be the total predicted influence based on ($\sum INR$).
Let $\sum_{i=0}^{n} C_i + \sum_{i=0}^{n} F_i = \sum INR$

$$CIM(v_i) = \sum_{i=0}^{n} C_i + \sum_{i=0}^{n} F_i \tag{7}$$

,which can be presented as a generalization of $C_{\mathbf{d}}$, where we consider the weight of $d_i^{in} \therefore d_i^{w(in)} = \sum_{i=0}^{n} C_i + \sum_{i=0}^{n} F_i$

$$CIM(v_i) = d_i^{w(in)} \tag{8}$$

It can be calculated from M in Equation (1),let $M[i,j] = w(e_{i,j})$, $\exists\ e_{i,j}$, $\therefore\ d_i^{w(in)} = \sum_{j=0}^{n} w(e_{i,j})$, where $i, j \in V$
For example, Let us consider SN shown in Figure 2 that includes Bob, Charlie, and Alice where Charlie and Alice interact with Bob, Bob interact with Charlie and Alice, and Alice interact with Bob, Where $INR = 5, 2, 1$ respectively. $CIM(v_{bob}) = 5$, $CIM(v_{Charlie}) = 2$, and

$CIM(v_{Alice}) = 1$. Let us calculate CIM of the entire SN.

$CIM = \frac{\Sigma^3{}_{i=1}[(5-5)+(5-2)+(5-1)]}{5\times(3-1)} = \frac{7}{10} = 0.7$. Then the centrality of graph is high, where Bob is the most influential user.

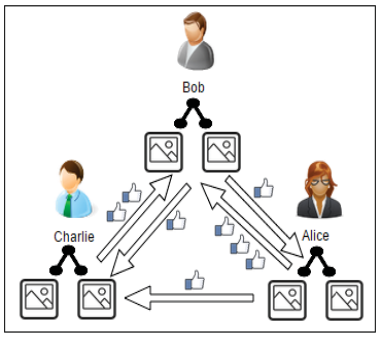


Fig. 2

EXAMPLE: A SOCIAL NETWORK OF THREE USERS BOB, CHARLIE, AND ALICE. WHERE THE USERS HAVE POSTED SEVERAL POSTS AND INTERACTED WITH EACH OTHER

5 Experiments

In order to assess the proposed approach, we perform two experiments on Flickr. The first experiment prove the robustness of CIM over centrality analysis, while the second experiment evaluate CIM in term of accuracy.

5.1 Dataset

We use the Flickr API to crawl data. We collect two datasets for the experiment and evaluation. We conduct our study on a Flickr Group. The group was selected based on:

(1) Activity volume,
(2) The number of Users, and
(3) Period of existence.

The Group has 14740 members and 5080 posted photos. The dataset includes data from 12/16/2004 to 12/30/2014. The dataset includes the users IDs, social interactions, and photos information(tags, and photo locations). After data preprocessing, the dataset has 7394 users, 523 contributing users, who shared 4415 P containing 25456 tags, and received 25459 social interactions. This data is illustrated in Table 3. The two main objectives behind this experiments are: (1)To prove the robustness of CIM, and (2)To evaluate the accuracy of CIM. We removed photos that received $INR < 2$; we also removed social interactions of users who made social interactions on their own photos. We also removed the social interactions between the contributing user and her followers if the social interactions are on the same posts and $INR > 2$ for both users to filter out possible conversations. However, this occurred few times. For the followers, only users who made $INR > 1$ are considered. We use this dataset to conduct our experiment to rank users and compare between the content and context-based influence measure. We build the ground-truth based on the observed social interactions from Flickr. The ground-truth includes the top 12 influential users' social interactions, where $INR = 8276$. The second dataset represents the tags and photo locations of the followers. Due to the fact that it takes a long time to crawl this part, we only collect the tags and locations of a maximum of 100 of their Photos. It takes a long time to crawl this part; Therefore, we select the top 12 influential users to crawl their followers' tags and locations. The second data set characteristics are shown in Table 4. We use this dataset to evaluate CIM by predicting the social interactions on the influential user's photos, and then compare between the predicted results and the ground-truth.

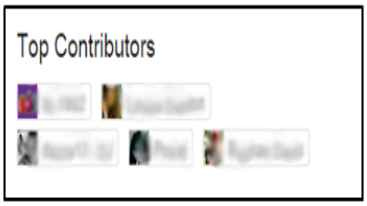


Fig. 3

FLICKR RANKS THE TOP 5 USERS BASED ON THE MOST POSTED PHOTOS IN THE GROUP CIM_{flickr}

Table 3

GROUP CHARACTERISTICS

Number of Users	Number of Photos	Number of Contributors	Number of Interactions	Number of Tags
7394	4415	523	25459	32877

Table 4

TOP 12 INFLUENTIAL USERS' FOLLOWERS' CHARACTERISTICS

Number Unique Followers	Number of Photos	Number of Unique Tags	Number of Unique Photos locations
3487	304134	580816	11366

5.2 Robustness of CIM

In order to prove the robustness of our approach, we apply CIM and the other approaches to the Flickr dataset. We rank the users based on the Content and Context-Based Influence measure. Based on the different measurements, we see that some users have not been ranked in some of the context-based approaches. We also observe that some of the top influential users were constantly ranked among the top rankings based on the different measures, which shows that those users have more impact.

To rank the users in our dataset, contributed users can at least have one follower to be in the ranking, which means that $d_i^{in} > 0$, and $\sum_{i=0}^{n} C_i + F_i > 0$ where these two variables are dependent. As d_i^{in} increases, $\sum_{i=0}^{n} C_i + \sum_{i=0}^{n} F_i$ also increases. INR also depends on content as well since without content users will have $INR = 0$. To compute CIM, we initialize the vector $C[v]$ and total influence, between line 1 and 2. We calculate the total $CIM(v_i)$ between line 3 and 8,if $\exists\ e_{i,j}$ between i and j. In line 11, we store each $CIM(v_i)$ of influential users in the vector $C[v_i]$. We can see that in each iteration $CIM[v_i]$ depends on d_i^{in}, as we can see in algorithm 2.

Then, we rank users using $C_{\mathbf{d}}$ where users who are ranked high based on how many followers they have. Secondly, we

rank users using $C_{\mathbf{b}}(v_i)$ which considers how much a user connect other users. We rank users using $C_{\mathbf{p}}(v_i)$ measure where it considers the importance the influential users' followers. As shown in Figure 3, Flickr also ranks the top 5 users based on a variation of CIM where it only considers the content. CIM_{flickr} rank users according to $Max\ \Sigma\#P$ in the group which can be calculated using $w(v_i)$. To mimic CIM_{flickr}, we rank the users in the dataset according to $CIM_{flickr}(v_i) = Max \sum_{i=0}^{n} p_i\ \forall$ i $\in V$. As mentioned before, SN is based on both content and context. In CIM, we consider both context and content. Where centrality analysis considers only SN structure and CIM_{flickr} considers only content. The rankings by $C_{\mathbf{b}}$ and $C_{\mathbf{p}}$ were not able to rank all the contributing users in the dataset, where $C_{\mathbf{d}}$ and CIM_{flickr} were able to rank all the contributing users in the dataset. However, $C_{\mathbf{d}}$ is based on context and CIM_{flickr} is based content, while CIM is based on both content and context.

Algorithm 2: Empirical Rank

Data: M: Weighted Adjacency Matrix, d_i^{in}: In-degree matrix
Result: $C[V]$:Ranked List

```
INR ← 0;
C[V] ← 0;
for all v_i ∑ M + 1 do
    for all v_j ∑ M + 1 do
        if d_i^in > 0 then
            if M[i, j] > 0 then
                INR ← INR+M[i, j] ;
            end
        end
    end
    C[V] ← INR
    INR ← 0
end
Return C[V];
```

We use Pearson's correlation coefficient to compare between CIM and the other rankings. Pearson's correlation coefficient is used to correlate two numerical attributes A and B. In our case, the two attributes are the two rankings, for example CIM and $C_{\mathbf{d}}$ (where $A = CIM$ and $B = C_{\mathbf{d}}, C_{\mathbf{p}}, C_{\mathbf{b}}, CIM_{flickr}$ respectively). It is computed as in Equation 9.

$$r_{A,B} = \frac{\sum_{i=0}^{n}(a_i b_i) - n\bar{A}\bar{B}}{n\sigma_A\sigma_B} \tag{9}$$

As shown in Figure 4, we observe a high correlation with $r_{A,B} = .95$ between CIM and $C_{\mathbf{d}}$, which makes sense, since CIC is a generalization of $C_{\mathbf{d}}$ where users with many number of unique followers can receive higher INR. We observe a low correlation between CIM and $C_{\mathbf{p}}$ with $r_{A,B} = .48$. This is because many users were not ranked in $C_{\mathbf{p}}$, however, correlating the top 25 influential users resulted in a high correlation between CIM and $C_{\mathbf{p}}$ with $r_{A,B} = .84$ that shows that top influential users also have important followers. We find a weak correlation between CIM and CIM_{flickr} with $r_{A,B} = .41$, which apparently shows that users who share many photos are not necessarily influential. We observe a good correlation with $r_{A,B} = .65$ between CIM and $C_{\mathbf{b}}$ which shows that influential users also connect many users together, where $C_{\mathbf{b}}$ fails to rank many users.

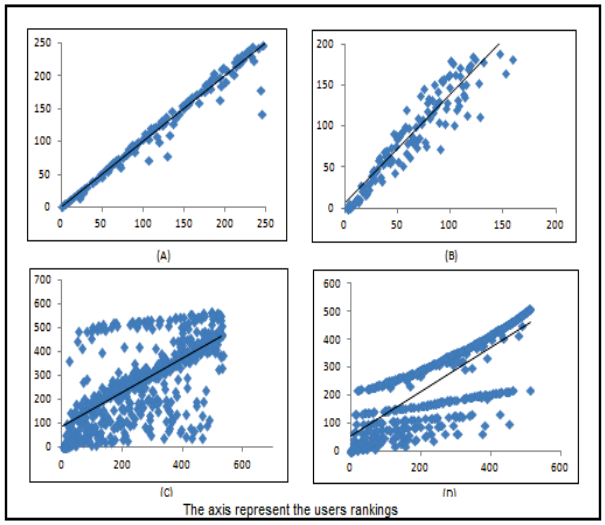


Fig. 4
(A)CORRELATION BETWEEN CIM AND $C_{\mathbf{D}}$ WITH $r_{A,B} = .95$ (B) CORRELATION BETWEEN CIM AND $C_{\mathbf{P}}$ WITH $r_{A,B} = .48$ (C) CORRELATION BETWEEN CIM AND $C_{\mathbf{B}}$ WITH $r_{A,B} = .65$(D)CORRELATION BETWEEN CIM AND CIM_{flickr} WITH $r_{A,B} = .41$ AXIS REPRESENT RANKING OF USERS

5.3 Accuracy of CIM

In order to evaluate the accuracy of CIM, we predict the social interactions for the top 12 influential users. Then compare the predicted social interactions to the ground-truth. We evaluate our approach using two features(F) to predict $\sum INR\ \forall$ influential users, where influential users $\in V$. We use the Positive Predictive Value (PPV) to evaluate the accuracy of our approach.

$$PPV = \frac{NumberofTruePositives}{NumberofPositiveCalls} \tag{10}$$

5.3.1 Using Tags Feature

In the first experiment, we select $F = (tags)$. We store the $tags$ of all P for each user i in a term frequency vector ($\vec{TF}$) where ($\vec{TF} = (p, tags)$) $\forall\ tags\ \forall\ P$, and $freq(p; tags)$ that represent the frequency of $\forall\ tags$. [10]

$$TF(d,u) := \begin{cases} 0 & if(p, tags) = 0 \\ 1 + log(1 + log(freq(p, tags))) & otherwise \end{cases} \tag{11}$$

Then we use Cosine Similarity Measure to find similarity between users to predict whether they would comment on or favorite the influential users' photos. The Cosine Similarity Measure measures the similarity between two users by considering the $x \cap y$ common tags and the uncommon tags $x \cup y$ and their frequency. The measure computes the cosine of the angle between two vectors. [10] Let $\vec{x} = \vec{TF}(v_i, tags)$ and $\vec{y} = \vec{TF}(v_i, tags)$. We want to calculate the similarity

$TSim$(influential user,follower) $\forall$ influential user, follower $\in V$. We compute $\frac{\vec{x}\vec{y}}{\|\vec{x}\|\|\vec{y}\|}$ where $\vec{x}\times\ \vec{y} = \vec{x} \cap \vec{y}$ and $\|\vec{x}\| \times \|\vec{y}\| = \vec{x} \cup \vec{y}\ \forall\ \vec{x},\vec{y} \in \vec{TF}$ We store the similarity $\forall$ followers of influential users in a similarity matrix(SM). SM is $P \times Followers$ as shown below. Each influential user is represented by her photos.

$$SMip := \begin{cases} SM - Tsim[i,p]) & if[i,p] \neq\ 0 \\ 1 + log(SM[i,p]) = 0 & if[i,p]\ =\ 0 \end{cases} \quad (12)$$

5.3.2 Using Tags and Photo Locations Features

In this experiment, we use $F = (tags, locations)$. We use $\vec{TF}(p, locations)$ to represent Photos' locations. We compute the similarity of users based on $locations$ between influential user and i by considering the frequency of the common locations $\vec{x} \cup \vec{y}$ and the uncommon locations $\vec{x} \cap \vec{y}$. $SLoc$(influential user, i) $= \frac{(x \cup y)}{(x \cap y)}$ $\forall$ influential users , $i \in V$ and $\forall\ x, y \in locations$. When using($F = 2$), the accuracy increases slightly. To compute the total similarity (Sim): Let Sim(influential user, i) $= \Omega \times TSim$(influential user, i)$+(1-\Omega) \times SLoc$(influential user, i) where Ω controls the relative importance between the two features. Table 5 and Figure 5 show the PPV results.

Table 5
. ACCURACY OF F=1 AND F=2

of Feature	PPV
F=1	.54
F=2	.56

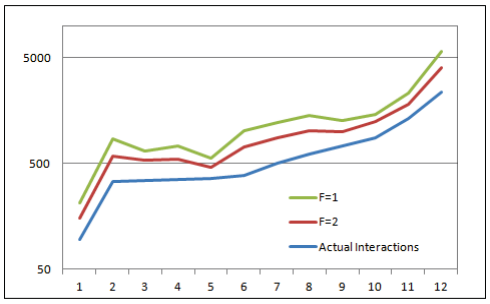


Fig. 5
ACTUAL VS. PREDICTED INR FOR EACH INFLUENTIAL USER USING F=1 AND F=2

6 Conclusion

We propose a Content-Based Influence Measurement that considers both context and content. We formally prove the centrality of CIM. We apply CIM and other measures to Flickr. We show that CIM proves robustness over other measurements. By correlating CIM and CIM_{flickr}, we show that CIM_{flickr} does not reflect the popularity of users in terms of INR. Therefore, considering content alone does not reflect influence. Moreover, Context-Based Measurements such as Centrality analysis measures only consider the SN structure, where as CIM uses the power of each user by considering content and context. We further evaluate the accuracy of CIM by predicting the social interactions. For our future work, we think that content is critical in making users influential, and users who dedicate their posts to a high-quality work, can become influential in time. We seek to explore different ways to evaluate the quality of content.

References

[1] Aggarwal, C. C. (2011). An introduction to social network data analytics (pp. 1-15). Springer US.

[2] Anger, I., Kittl, C. (2011, September). Measuring influence on Twitter. In Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies (p. 31). ACM. Chicago

[3] Bakshy, E., Hofman, J. M., Mason, W. A., & Watts, D. J. (2011, February). Everyone's an influencer: quantifying influence on twitter. In Proceedings of the fourth ACM international conference on Web search and data mining (pp. 65-74). ACM.

[4] Brandes, U. (2001). A faster algorithm for betweenness centrality*. Journal of Mathematical Sociology, 25(2), 163-177

[5] Cha, M., Haddadi, H., Benevenuto, F., & Gummadi, P. K. (2010). Measuring User Influence in Twitter: The Million Follower Fallacy. ICWSM, 10, 10-17.

[6] Community Detection and Mining in Social Media Tang, Lei; Liu, Huan (2010-05-05). Community Detection and Mining in Social Media (Kindle Location 2). Morgan & Claypool Publishers.

[7] Freeman, L. C. (1979). Centrality in social networks conceptual clarification.Social networks, 1(3), 215-239

[8] Granovetter, M. S. (1973). The strength of weak ties. American journal of sociology, 1360-1380

[9] Ghosh, R., & Lerman, K. (2010). Predicting influential users in online social networks. arXiv preprint arXiv:1005.4882.

[10] Han, J., Kamber, M., & Pei, J. (2006). Data mining, southeast asia edition: Concepts and techniques. Morgan kaufmann.

[11] Katz, E., & Lazarsfeld, P. F. (1970). Personal Influence, The part played by people in the flow of mass communications. Transaction Publishers.

[12] Lee, C., Kwak, H., Lee, C., Park, H., & Moon, S. (2010, April). What is Twitter, a social network or a news media?. In Proceedings of the 19th international conference on World wide web (pp. 591-600). ACM

[13] Lee, C., Kwak, H., Park, H., & Moon, S. (2010, April). Finding influentials based on the temporal order of information adoption in twitter. In Proceedings of the 19th international conference on World wide web (pp. 1137-1138). ACM.

[14] Li H, Cui JT, Ma JF. Social influence study in online networks: A three-level review. Journal of Computer Science and Technology 30(1): 184–199 Jan. 2015. DOI 10.1007/s11390-015-1512-7

[15] Li, J., Peng, W., Li, T., & Sun, T. (2013). Social network user influence dynamics prediction. In Web Technologies and Applications (pp. 310-322). Springer Berlin Heidelberg.

[16] Leavitt, A., Burchard, E., Fisher, D., & Gilbert, S. (2009). The influentials: New approaches for analyzing influence on twitter. Web Ecology Project, 4(2), 1-18

[17] Newman, M. E. (2001). Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality. Physical review E, 64(1), 016132

[18] Probst, F., Grosswiele, D. K. L., & Pfleger, D. K. R.(2013). Who will lead and who will follow: Identifying Influential Users in Online Social Networks.Business & Information Systems Engineering, 5(3), 179-193

[19] Reilly, C. F., Salinas, D., & Leon, D. D. (2014, March). Ranking Users Based on Influence in a Directional Social Network. In Computational Science and Computational Intelligence (CSCI), 2014 International Conference on (Vol. 2, pp. 237-240). IEEE.

[20] Sang, J., & Xu, C. (2013). Social influence analysis and application on multimedia sharing websites. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP), 9(1s), 53.

[21] Sun, B., & Ng, V. T. (2013). Identifying influential users by their postings in social networks. In Ubiquitous Social Media Analysis (pp. 128-151). Springer Berlin Heidelberg.

[22] Weng, J., Lim, E. P., Jiang, J., & He, Q. (2010, February). Twitterrank: finding topic-sensitive influential twitterers. In Proceedings of the third ACM international conference on Web search and data mining (pp. 261-270). ACM.

[23] Zafarani, R., Abbasi, M. A., Liu, H. (2014). Social media mining: an introduction. Cambridge University Press.

Big Data Analytics on Acquisition Issues of Public Healthcare

A. Haibo Wang [1], B. Chenhui Zhou [1], C. Wei Wang [1] and D. Yaquan Xu [2]
[1]Sanchez School of Business, Texas A&M International University, Laredo, Texas, USA
[2]School of Science and Technology, Georgia Gwinnett College, Lawrenceville, Georgia, USA

Abstract -*This study aims to analyze the existing federal acquisition policy in public healthcare in the United States, especially in the case of childhood immunization, and present a framework on developing a comprehensive system to analyze data on public healthcare using an open source-based meta-data acquisition and aggregation tool based on Hadoop, MapReduce and MongoDB to integrate databases from different agencies. We design data-driven XML schema for tagging the data entries and implement a package using open source software R to aggregate XML-transformed data into time and space dimensions, then propose a new multiple items multiple vendors model to analyze the data, with the objectives of lowering costs including transportation and wastage. This study uses the case of children's immunization vaccines as an example to illustrate its contribution. . The proposed system also addresses the challenge of implementing the new government recommended immunization requirements in a cost effective manner based on the data from different sources. In addition, the new acquisition decision support system enhances the transparency of healthcare costs, especially in the public health service.*

Keywords: Big Data Analytics; Public Healthcare; Data Envelopment Analysis; Multi Criteria Decision Analysis

1 Introduction

The shortage of the influenza A virus (H1N1) vaccine in 2009 and other similar incidents in the past ten years[1], [2], including shortages in children's immunization vaccines, is what motivates the authors of this study to investigate the existing acquisition policy in the healthcare industry. The frequent wastagein vaccine[3]and other emergency medical supplies cost taxpayers billions of dollars each year due to a poorly managed acquisition policy by the healthcare industry[4]. This wastage adds to the high cost of healthcare in the United States, which is higher comparing to other developed countries[5]. One major area where vaccine wastage occurs is during childhood immunizations, which is compounded by the challenge of incorporating new vaccines into the already crowded immunization schedule adopted by the U.S. government. In addition, an increasing numbers of health providers and parents are opposed to add immunization requirements, which prevents an efficient resource management [6]-[8].

One approach to reduce wastage and overall costs in the healthcare industry is to develop better acquisition policy tools than are currently available to decision makers. According to the Centers for Disease Control and Prevention (CDC), there are a number of manufacturers who can provide multiple vaccines and other emergency supplies with different pricing schedules, even under the Federal contract purchase price due to different purchase financing systems[9]-[12]. These manufacturers can also offer different pricing schedules on the purchase of combinations of vaccines to enable purchasers to take advantage of the economies of scale[13]. However, these manufacturers are privately owned and profit driven organizations, and their production schedule depends on the nature of vaccine production and on accurately predicting the market needs. These needs include purchase contracts from the Federal government, public sector and private clinics, which result in manufacturers having little motivation to provide pricing flexibility[14], [15]. In addition, the data on vaccine production and demand is not shared among the stakeholders of public healthcare and different systems using by different agencies cannot integrate the data smoothly. These challenging issues are here investigated at first. We present a framework on developing a comprehensive system to analyze data on public healthcare using an open source-based meta-data acquisition and aggregation tool based on Hadoop, MapReduce and MongoDB to integrate databases from different agencies. In the meantime, we design a data-driven XML schema for tagging the data entries and implement a package using open source software R to aggregate XML-transformed data into time and space dimensions, then a new data-driven acquisition model is proposed to lower costs and reduce wastage.

2 Prior policy analysis and studies

According to a 2006 study by [16], none of the existing decision support systems which handle contract types and organization relationships consider quality management because its complexity, although the U.S. Department of Defense began employing quality management techniques after the WWII[7], [17], [18]. Other government agencies such as NASA also employ a quality assurance standard to their contractors[19]. Quality management concepts have also been examined in the

context of the public health care field[20], [21]. In the private sector, a quality management decision support system is implemented as a part of the Enterprise Resource Planning System as a tool to collect data from different suppliers[22]. Academic researchers in the quality management fields study the impact of information on contracting and quality management[23] and some recommend the structure of a decision support system as a concept[24], [25] and define a prototype but never implemented it as operational tools[26] that can be easily used/integrated into existing procurement systems. Their proposed mathematical models focus on finding the key factors by using analytic hierarchy processes for multiple criteria under the assumption of independency of criteria[27]. There is no quantitative analysis tool available to a decision-maker in the acquisition process to evaluate/select vendors. Many existing decision support systems in the acquisition process are document based to provide guidelines/information, for example, the Surface Decision Support System provided to the U.S. Department of Transportation[28].

According to the General Service Administration and the U.S. Department of Health and Human Service, the existing acquisition policies and procedures conform to the Federal Acquisition Regulation System (FARS), which monitors contracting relationships between the contracting offices and contractors. A quality surveillance plan should be implemented and the contractor should be evaluated on performance based criteria. There are 11 criteria to be considered but the evaluation is carried out by an expert panel with a list of weighted factors to be considered. There is no general decision support system available to the decision maker (expert panel) on evaluating the vendors based on the quality management criteria in addition to the criteria listed by the FARS. These criteria are weighted according to the general guideline but no specific model or tool is available to integrate the criteria into a software system for decision support. The FARS and its related policies might bring challenges in adopting this decision support system software across federal agencies. One of the main challenges will be the organizational resistance to adopt a new decision tool and the difficulty of integrating that tool into the different systems each agency uses. However, there is no barrier in the private sector for adopting the proposed system as an analytical tool to help businesses improve their performance. Contractors can also employ the tool to design financial strategies to customize service packages, or to better estimate the cost of transportation. If the decision support system becomes popular among the contractors, it will increase its credibility among federal agencies and encourage the effort to integrate it into their decision support systems. By designing this new tool in a spreadsheet based format, this should reduce implementation barriers, as it can be easily integrated into existing procurement systems.

Currently, The Center for Disease Control and Prevention (CDC), Department of Health and Human Service and other government agencies use a commercial version of a relational database such as ORACLE to store the public healthcare information. For structured and clear data prepared by professional consultants or IT department, ORACLE is the most popular commercial database management software worldwide. Nowadays, most data is collected in real time from in-memory data collecting systems such as scanner (RFID or bar code), mobile devices, and other electronic devices. The relational database software such as ORACLE is not designed to meet the challenges of data generated from internet and mobile devices in term of scale and agility and application developers have to solve the mismatch issues between the in-memory data structure of big data and the underlying structure of relational database systems by converting the in-memory data to tables and other relational structures.

3 Methodology

This study departs from current acquisition practices and brings significant innovation to the data-driven acquisition model. Its main advantages and innovations are: 1) a comprehensive system to analyze data on public healthcare using an open source-based meta-data acquisition and aggregation tool; 2) data-driven XML schema for tagging the data entries and aggregate XML-transformed data into time and space dimensions 3) a new two-step decision model with a data envelopment analysis (DEA) approach and a non-linear multiple criteria decision analysis (MCDA) approach; 4) an integrated spreadsheet based software system that includes the cost of distribution and transportation, and is designed to be easily implemented with current medical management systems in public and private sectors.

The tools and methods used for the big data analytics in this study include Hadoop, MapReduce and MongoDB, XMLschema, and R software (See Fig. 1). These open source software provide a number of benefits including free distribution, low cost, worldwide developer community, and are easy to use. Hadoop is an open source software environment for large data storage and distribution that uses low cost computer clusters and MapReduce is an open source functional programming environment for data manipulation and processing. Both Hadoop and MapReduce can be deployed in the cloud or a local datacenter. MongoDB is an open source cross-platform NoSQL (not only SQL) database management software that provides the benefit of improved system productivity through an application oriented data structure and also improves the data utilization efficiency for storing and processing large data volumes, reducing latency and improving input/output performance. XML schema is the widely used object management standard and the foundation of semantic systems such as search engines and artificial intelligence. R software is an open source cross-platform programming language with a platform adaptive run-time environment for statistical computing and graphics. There are more than 3,500 application/function based packages developed by worldwide researchers and widely used for a

variety of statistical models including statistical tests, linear and non-linear models, time series forecasting, data mining and predictive analytics, etc. There more functions and algorithms in R software than in the other three most popular commercial statistical software combined. It provides a collaborative environment for new applications or functions to be added on as a cross-platform library.

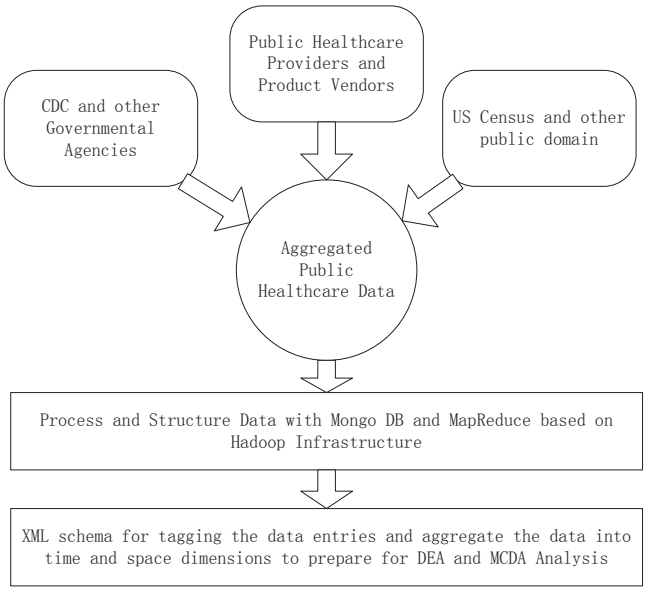


Fig. 1. Framework on Data Acquisition and Aggregation on Public Healthcare

The model and methods are programmed into a speadsheet based software system. The real world data such as the federal contracted companies and other medical suppliers from the CDC website (see Fig. 2) and other public domains will be collected using web based free screen scraper or crawler software and used to evaluate the performance of the proposed model (See Fig. 3). The outcomes of the proposed model and methodology are measured through a simulation using real world data with different scenarios in terms of acquisition cost, quality management of different vendors, and other criteria.

The first stage of the research is to analyze the data collected and stored in a spreadsheet format by the web based automatic scraper/crawler from the CDC websites, U.S. Census, the U.S. Department of Health and Human Services, General Service Administration, and other public domains. The development of the demand forecast involves several statistical methods (available on the spreadsheet software) such as multiple regression analysis, nonlinear regression, decomposition analysis, adaptive filtering, and trend analysis (See Fig. 4). The forecasting error is analyzed through the analysis of variance.

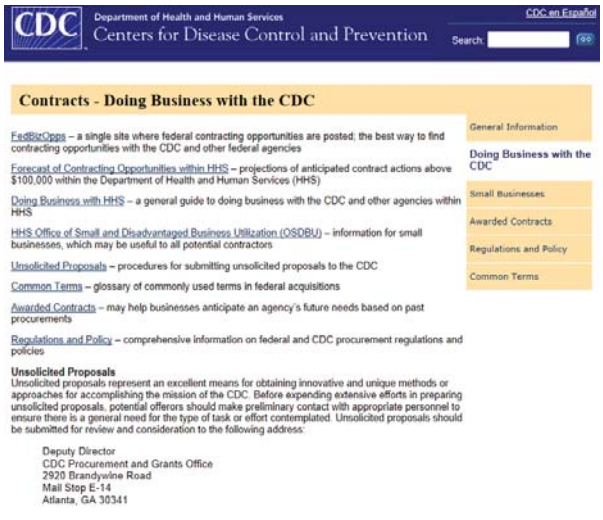


Fig. 2. CDC suppliers and contractors data sources

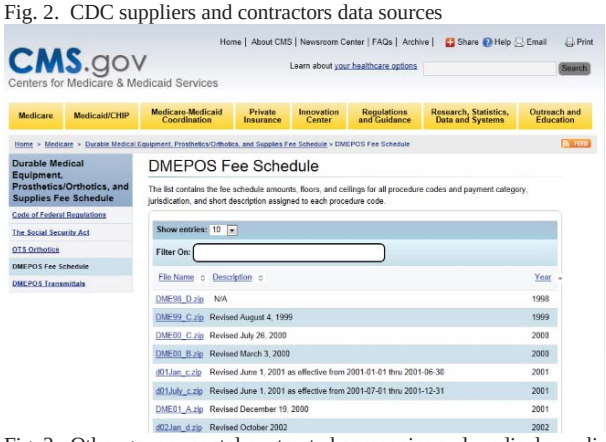


Fig. 3. Other governmental contracted companies and medical suppliers

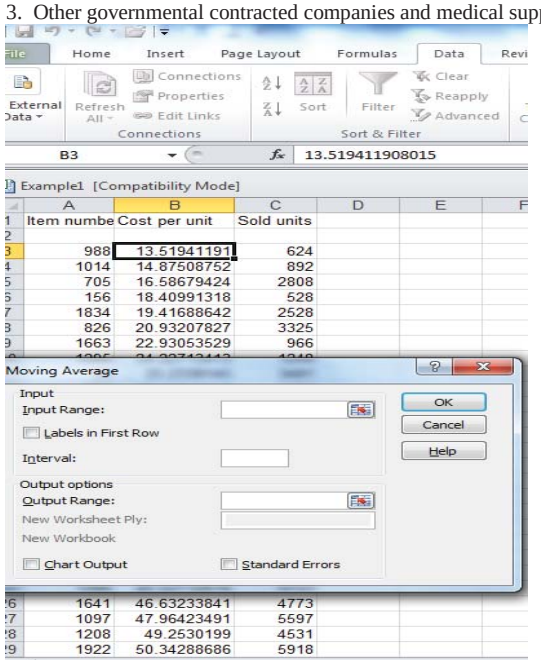


Fig. 4. Forecasting demand using spreadsheet

The second stage is to build a decision model for acquisition, which consists of two steps: 1) evaluation and selection; and 2) cost optimization. The *evaluation process* ofmultiple suppliers is based on an extension of the data

envelopment analysis (DEA) model (see Fig. 5) using quality management criteria in addition to the criteria listed in the Federal Acquisition Regulation for the public sector and reports the rank of the suppliers of each product to create a preferred supplier pool. In the private sector, the criteria listed in the FARS will be considered as optional criteria. The *selection process* of multiple suppliers is a nonlinear multi criteria decision analysis (MCDA) model (see Fig. 6) based on the analysis of the interdependence among products and suppliers from the pool of preferred suppliers and it reports the best groups of suppliers. The transportation *cost optimization* is based on the fixed-charge transportation model.

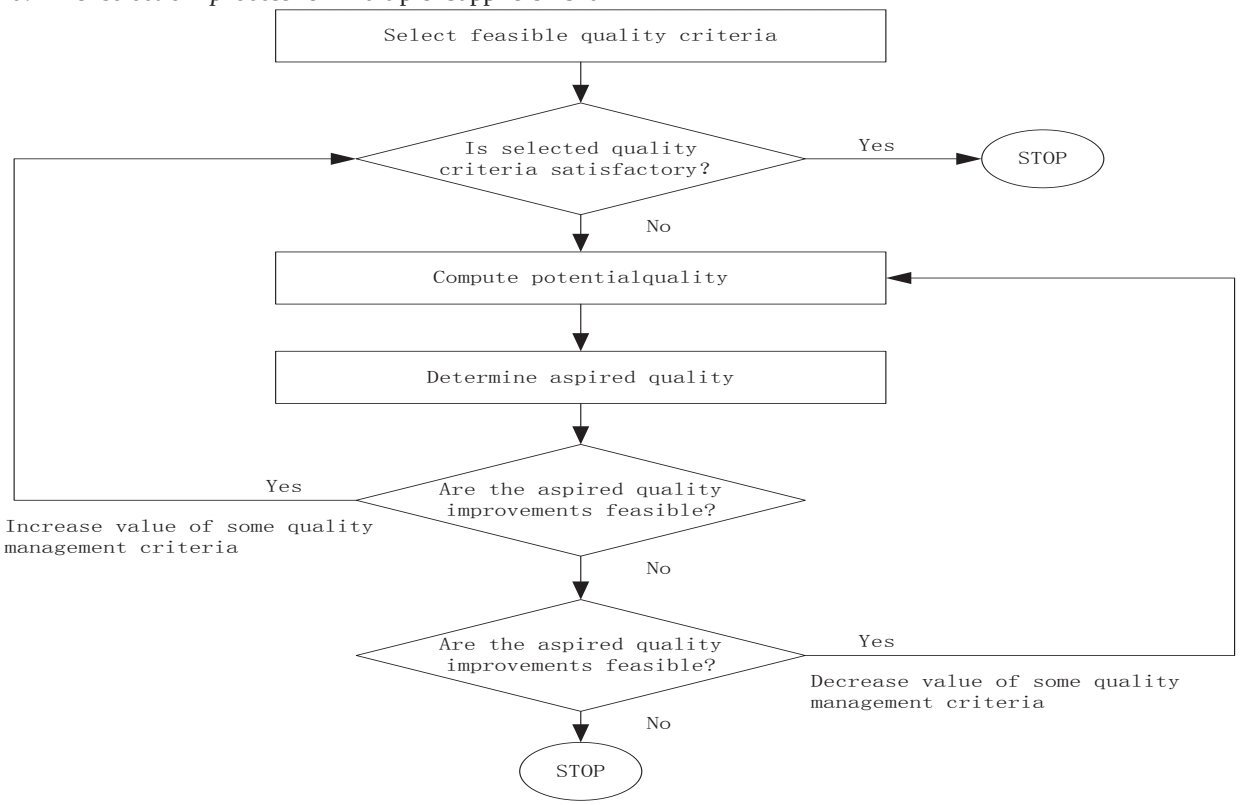


Fig. 5. DEA model use quality management criteria

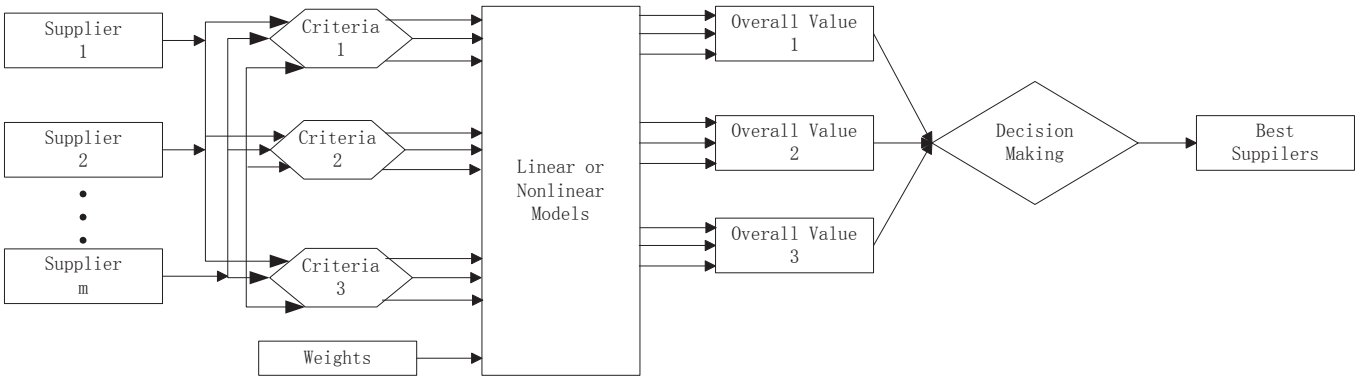


Fig. 6. MCDA model for supplier selection procedure

The quality management based DEA model provides a rounded judgment on supplier performance taking into consideration multiple quality management criteria simultaneously, including the criteria listed in the FARS and then combining them into a single measure for quality control. The mathematical model is solved for every medical item and the relative quality score of each supplier is determined (see Figure 8). The results of the DEA model show that the higher a supplier's quality score in relation to the corresponding score of another supplier, the higher the rank of this supplier is in term of quality. This interdependent relationship plays an important role in terms of supply risk and long term partnership. The DEA model is an effective approach for vendor evaluation but it can not be used to evaluate the supply risk and partnership during the selection process. This important issue is often overlooked in the evaluation of

vendors, especially when they are contracted by the government for longer terms and it can cause shortages and waste of vaccines.

In the nonlinear MCDA approach, a decision-maker could simultaneously consider a pool of suppliers from the evaluation process and also choose a set of suppliers to address supply risks and partnership issues. The basic notation of this process is defined as:

$N = \{v^1, \ldots, v^n\}$ is the set of suppliers

$P = \{p^1, \ldots, p^m\}$ is the set of medical items

$v^i, (i = 1, \ldots, n)$ is supplier i

$p^j, \quad (j = 1, \ldots, m)$ is medical item j

$I(v^i)$ is the number of medical items provided by supplier v^i in the selection pool

$Q = \{1, \ldots, q\}$ is the set of interdependent criteria

c_k^i is the effect of supplier i on criterion k

S_k is a set of suppliers that if selected together have some positive or negative effect on criterion k

$\gamma(S_k)$ is the amount of effect (positive or negative) of an interacting set S_k on criterion k

w_k is the weight associated with criterion k

$\phi(S)$ is the total payoff benefit of a subset of suppliers $S \subseteq N$

D is the number of suppliers selected

X_{ij} is equal to 1 if an supplier v^i is selected for item j and 0 otherwise

$y_i = \frac{\sum_{j=1}^{m} x_{ij}}{I(v^i)}$ is equal to 1 if all of items from supplier v^i are selected as a bundle of items and 0 otherwise.

In general, the interdependency relationship can be measured by a number of criteria: economies of scale, resource and risk sharing, and ownership or partnership. With regard to the interdependent relationship, there is a weight to each criterion, and the model is represented by a nonlinear programming function as follows:

$$Max \sum_{\substack{i \in N \\ j \in P}} x_{ij} \left[\sum_{k \in Q} w_k c_k^i \right] + \sum_{k \in Q} \sum_{s_k \subseteq N} \left(w_k [\gamma(S_k)] [\sum_{i \epsilon s_k} c_k^i] \right) \prod_{i \epsilon S_k} x_{ij} \quad (1)$$

s.t.

$$\sum_{i \in N} y_i \leq D \quad (2)$$

$$x_{ij}, y_i \in \{0,1\}, i = 1, \ldots, n, j = 1, \ldots, m$$

This function can be solved by a spreadsheet solver such as Frontline Premium Solver and provide the selection results for vendors in the next step of cost optimization. In the case of children's immunization problems, the schedule of combined vaccines will be included in the interdependent criteria in the decision model. The transportation cost optimization model is based on the fixed-charge transportation problem, which arises frequently in the application areas of scheduling and cost control. In a medical supply system, a variety of sources including international suppliers, local manufacturers, and wholesalers are all capable of sending medical supplies to a primary distributor. The primary distributor has to develop a transportation strategy for cost control.

The next step in the process of solving this issue is to model it as a fixed-charge transportation problem. The fixed charge problem determines the amount of shipments made from a given set of suppliers to a single destination, such that the total demand is satisfied at a minimum cost. A fixed charge and cost proportional to the quantity shipped occurs when a supplier is employed. The fixed charge problem can be mathematically formulated in terms of an integer programming problem as follows:

$$Min\ TC = \sum_{i=1}^{M} (f_i d_i + F_i z_i) \quad (3)$$

s.t.

$$\sum_{i=1}^{M} d_i = D \quad (4)$$

$$0 \leq d_i \leq b_i z_i$$

$$z_i \in \{0,1\}\ and\ i = 1, \ldots, M$$

Where D is the demand magnitude, M is the number of suppliers, F_i is the fixed management cost incurred when using supplier i, b_i is the capacity of supplier i, f_i is the elective unit variable cost of supplier i, and z_i is an integer variable taking the value of 1 when supplier i is used, and 0 otherwise. The solution to the problem is a vector $d=(d_1..d_2..d_M)$ of nonnegative integers (also called the transportation profile) for which d_i is the amount allocated to supplier i.

This linear integer programming model can be solved using a standard spreadsheet solver and reported to the decision maker in a numeric or graphic fashion.

4 Conclusion

While the analytical methods in this study are ready to be implemented, the collection of data from different sources may be limited. Different governmental agencies have their own lists of vendors and specific evaluation criteria/procedures under the Federal Acquisition Regulation System. In the case of childhood immunization, the list of vendors of vaccines and their cost profiles might not be frequently updated.

This study departs entirely from current acquisition practices and brings significant innovation to the current acquisition model. The proposed framework also addresses the challenge of implementing the new government

recommended immunization requirements in a cost effective manner.

5 References

[1] A. T. Chamberlain, K. Wells, K. Seib, A. Kudis, C. Hannan, W. A. Orenstein, et al., "Lessons Learned From the 2007 to 2009 Haemophilus influenzae Type B Vaccine Shortage: Implications for Future Vaccine Shortages and Public Health Preparedness," Journal of Public Health Management and Practice, vol. 18, pp. E9-E16 10.1097/PHH.0b013e31821dce27, 2012.

[2] A. R. Hinman, W. A. Orenstein, J. M. Santoli, L. E. Rodewald, and S. L. Cochi, "VACCINE SHORTAGES: History, Impact, and Prospects for the Future*," Annual Review of Public Health, vol. 27, pp. 235-259, 2006.

[3] S. Setia, H. Mainzer, M. L. Washington, G. Coil, R. Snyder, and B. G. Weniger, "Frequency and causes of vaccine wastage," Vaccine, vol. 20, pp. 1148-1156, 1/15/ 2002.

[4] D. Levinson, "Preventing Health Care Fraud: New Tools and Approaches to Combat Old Challenges. Committee on Finance," U. S. Senate, Ed., ed, 2011.

[5] World Health Organization, "World Health Statistics," 2011.

[6] L. J. Fagnan, S. A. Shipman, J. A. Gaudino, J. Mahler, A. L. Sussman, and J. Holub, "To Give or Not to Give: Approaches to Early Childhood Immunization Delivery in Oregon Rural Primary Care Practices," The Journal of Rural Health, vol. 27, pp. 385-393, 2011.

[7] S. G. Robison, H. Groom, and C. Young, "Frequency of Alternative Immunization Schedule Use in a Metropolitan Area," Pediatrics, vol. 130, pp. 32-38, July 1, 2012 2012.

[8] H. Safi, J. G. Wheeler, G. R. Reeve, E. Ochoa, J. R. Romero, R. Hopkins, et al., "Vaccine Policy and Arkansas Childhood Immunization Exemptions: A Multi-Year Review," American Journal of Preventive Medicine, vol. 42, pp. 602-605, 2012.

[9] M. S. Coleman, N. Sangrujee, Z. Fangjun, and C. Susan, "Factors Affecting U.S. Manufacturers' Decisions To Produce Vaccines," Health Affairs, vol. 24, pp. 635-642, 2005.

[10] W. Orenstein, "Protecting our Kids: What is causing the current shortage in childhood vaccines? ," C. o. G. Affairs, Ed., ed, June 10, 2002.

[11] L. E. Rodewald, W. A. Orenstein, D. D. Mason, and S. L. Cochi, "Vaccine Supply Problems: A Perspective of the Centers for Disease Control and Prevention," Clinical Infectious Diseases, vol. 42, pp. S104-S110, March 1, 2006 2006.

[12] G. Freed and A. Cowen, "State-Level Perspectives on Vaccine Purchase Financing," 2002.

[13] A. K. Shen, L. E. Rodewald, and G. S. Birkhead, "Perspective of Vaccine Manufacturers on Financing Pediatric and Adolescent Vaccines in the United States," Pediatrics, vol. 124, pp. S540-S547, December 2009 2009.

[14] J. E. Russo, M. G. Meloy, and T. J. Wilks, "Predecisional Distortion of Information by Auditors and Salespersons," Management Science, vol. 46, p. 13, 2000.

[15] J. Peckenpaugh, "New Rules Should Make Competition Routine in Government, Says OMB," ed: GovExec.com, 2003.

[16] R. F. Shehane, "A framework for knowledge-aware service contract quality management decision support systems," 3230357 Ph.D., Nova Southeastern University, United States -- Florida, 2006.

[17] G. A. Pitman, J. G. Motwani, and D. Schliker, "Total quality management in the American defence industry - A case study," The International Journal of Quality & Reliability Management, vol. 11, pp. 101-101, 1994.

[18] M. Goh and G.-H. Tay, "Implementing a quality maintenance system in a military organization," The International Journal of Quality & Reliability Management, vol. 12, pp. 26-26, 1995.

[19] NASA Office of Inspector General, "Review of Performance-Based Service Contract Quality Assurance Surveillance Plans," 2002.

[20] B. A, "Specifying Quality in Health Care," Journal of Management in medicine, vol. 8, pp. 5-8, 1994.

[21] D. WE, The New Economics for Industry, Government, Education, 2nd ed. Cambridge, Massachusetts, USA: The MIT Press, 1994.

[22] F. Shafiei and D. Sundaram, "Multi-Enterprise Collaborative Enterprise Resource Planning and Decision Support Systems," in Proceedings of the 37th Hawaii International Conference on System Sciences, Big Island, Hawaii, 2004.

[23] S. Baiman, P. E. Fischer, and M. V. Rajan, "Information, Contracting, and Quality Costs," Management Science, vol. 46, p. 776, 2000.

[24] M. Corbett, The Outsourcing Revolutiont. Chicago, IL, 2004.

[25] M. Greaver, Strategic Outsourcing. New York, New York, USA: AMA Publication, 1999.

[26] R. Bose, "Knowledge management-enabled health care management systems: capabilities, infrastructure, and decision-support," Expert Systems with Applications, vol. 24, pp. 59-71, 1// 2003.

[27] C. Holsapple and A. Whinston, Decision Support Systems: A Knowledge-Based Approach. Saint Paul, Minnesota: West Publishing Company, 1996.

[28] Federal Business Opportunity. Available: https://www.fbo.gov

A research of rule discovery method for process object based on large scale data

Tao Du , Shouning Qu
School of Information Science and Engineering
University of Jinan
Jinan, China
eo_dut@ujn.edu.cn

Yushui Geng, Xingang Wang
School of Information
Qilu University of Technology
Jinan, China
wxg@qlu.edu.cn

***Abstract*—a model of process object is proposed in this paper based on the large scale data generated in process industry. According to the characteristics of process object model, a new method of rule discovery method for process object based on large scale history data is proposed. This method named T-C-A-C/T (Time series-Clustering-Association-Chain/Tree Flow) can be divided into several steps as: the raw data set will be adjusted by matching time series according to monitoring point in each link of process industry; then based on the adjusted data set, all monitoring points will be clustered, and the clustering result will be calculated to obtain 2-itemsets association rules; then these 2-itemsets rules will be used to establish the association-chain; at last, by computing each link in process object, the rule of producing states will be discovered and the knowledge base will be created. This method has been applied in some production processes of thermal power, and good effect has been achieved.**

Keywords—process object; time series; tssociation rule; state chain

I. INTRODUCTION

With the development of industrial control technology, distributed control system has been one of most important application systems in industrial production. Meanwhile with the computer, networks and database technology progress, a large number of data has been accumulated in management information system, so how to discover more important content in these data than query and statistics has been an attractive research field. In industrial production, the demand of automatically optimizing industry process and intelligent making control decision by knowledge base has been strong, but the large scale of history data brings difficulties as the cost of dealing with them, and many noisy data would be mixed in normal data. In1980s, the technology of Data Mining and Knowledge Discovery in Database were born with information technology development, and now they are the focus research fields [1, 2]. By intelligent computing technology, quick and acute knowledge should be provided to users, and the inner and important information should be discovered to help user make decision, which are main functions of data mining. But in distributed control system, the data set accumulated in process industrial object has characteristics as: the parameters of producing procedure are complex; the relevance of different link is very strong; the changing trend of data is nonlinear; and the sequence of data time series do not conform to real links in industry [3]. And in most process industrial procedure, the integrity and real-time are emphasized: real-time means the data should be captured in time; integrity means the complex parameter coupling and mutual restriction relationship. Based on these characteristics, single optimized method in process control can't effectively improve industrial parameters and control strategy in detail for the lack of data evidence. In this paper, the process object would be researched entirely to obtain the targets of discovery the rule of process object operating states in large scale of history data and using these rules. The rest of this paper is organized as: in section 2, the key problems of knowledge discovery in process object are introduced; in section 3, the related works are analyzed in detail; in section 4, the algorithm named T-C-A-C/T is proposed; in section 4, the algorithm is proved by experiment with real producing data set; at last the conclusion of this paper is summarized.

II. THE DEFINITION OF PROCESS OBJECT

In process industrial production, the whole procedure is consisted of many links according to different requirements, and every link should be added to detection devices, and the captured data would be stored in data storage center by interface to monitor the producing state. In fig1, the structure of collecting producing data in process industry is described.

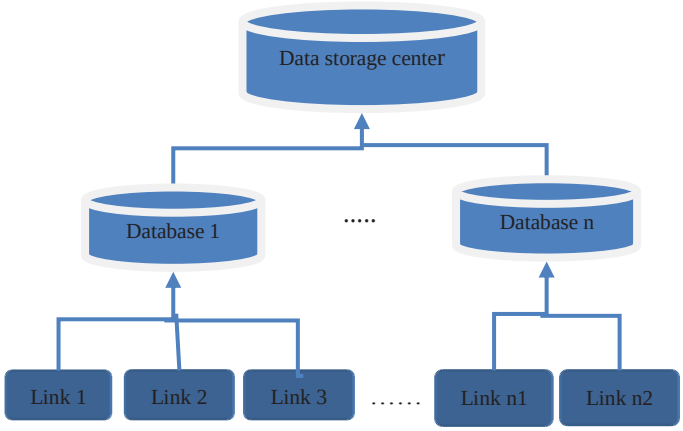


Fig. 1. The structure of process object

According to figure 1, the example of captured data stored in data storage center can be described as table1. From fig1 and table1, it can be concluded that the links of process industrial have transitivity, which leads the state data has transitivity too. According to the transitivity, a series of rule chains can be established, and using these rule chains, a knowledge base of process object can be created to assistant make decision. How to extract knowledge from large scale of data accumulated in process object is the focus of this paper, and the problems should be resolved are summarized as: the time series of links

should be discovery for the defect of data collecting technology, the order of links are not accord with the industrial order in process; the data set should be induced and classified, for the raw data set is too large that the efficiency of calculation is too low; for the relationship of process object has multi links characteristic, efficiently extracting rule chain from process object data is the key problem and the difficulty of research; at last, the expression and application of rules should be researched to help make decision in process industrial control.

TABLE I. THE DATA OF EACH LINK

detection time	Link 1	Link 2	Link 3	…	Link n-1	Link n
T_1	X_{11}	X_{21}	X_{31}	…	$X_{n-1,1}$	X_{n1}
T_2	X_{12}	X_{22}	X_{32}	…	$X_{n-1,2}$	X_{n2}
T_3	X_{13}	X_{23}	X_{33}	…	$X_{n-1,3}$	X_{n3}
…	…	…	…	…	…	…
T_k	X_{1k}	X_{2k}	X_{3k}	…	$X_{n-1,k}$	X_{nk}

III. EXISTING RESEARCHES

Process object data has obvious features of large scale of data as 3V: Volume, Velocity, and Variety, for the importance of process industry in national economy, many researchers have focused on this field, and many methods have been proposed to improve the performance of using history data.

At first, the analysis of Time Series is widely applied in finance, commerce, medical treatment, and weather prediction etc. The technology of dealing with time series data has also been improved, and many methods have been proposed. In ref[4], a mixed model named FTSGA is designed based on time series, and its performance is ensured by applying genetic algorithm to improve the efficiency of data retrieval; in this model, time series would be well arranged, but its computational complexity is too high. In ref [5], M.Gas adopts the sliding window method to discrete the time series data and then extract rules, and in this method, the rules can be discovered quickly, but their time series are not acute.

To improve the efficiency of discovery rules, raw data set should be induced and classified, and many methods have been proposed. In ref [6], Marina Resta discoveries association rules based on the method of SOMs, in which the raw data set would be classified by multi-distance calculation; in this method, the standard of classifying is the key of influencing efficiency. In ref [7], Imran Khan discovers rule based on artificial neural networks, and the raw data would be compressed by pruning strategy; in this method, the amount of calculation would be increased rapidly with data increasing. In ref [8], Erol Egrioglu proposes a new method of fuzzy time series based on particle swarm optimization algorithm, and the relationship of raw data set is estimated by a fuzzy matrix and would be compressed by their relationship; in this method, the efficiency of compression is relative high, but the accuracy is not satisfied for the fuzzy relationship.

Among existing discovery relation association researches, classical Apriori algorithm is still the most popular one [9]. In Apriori, rules would be extracted from frequent patterns which are generated from candidate frequent item sets; the main defect of Apriori is the calculation efficiency is too low, and when data increasing, the calculation would increase exponentially. Then many improved algorithms were proposed to improve the calculation efficiency. FP-growth algorithm [10] adopts tree-based structure to judge frequent pattern instead of candidate frequent item sets; in this algorithm, the efficiency is obvious improved, but the procedure is designed too complex and the memory cost is high. CHARM (Closed Association Rule Mining) [11] adopts vertical structure to discovery frequent items instead of horizontal structure, and this algorithm is easy to design but the efficiency is not as high as FP-growth.

Besides on the technologies mentioned above, the knowledge expression is also one important technology making the result of discovery rules can be used by user. But existing researches are most designed for special application, and there are not fit methods to express the knowledge of process object.

Based on the former analysis, it can be concluded that the technology of deal with large scale data of process object is still not perfect, and especially in knowledge expression, so in this paper, the method of improving the capability of dealing with large scale data in process object.

IV. THE DESIGN OF T-C-A-C/T

A. The structure of T-C-A-C/T

In T-C-A-C/T, the procedure can be divided into: data sampling, arrangement of time series, classifying the data set, clustering the data set, discovery association rules, creating rules' chain, and establishing knowledge base. The procedure of T-C-A-C/T is introduced in detail in section B.

B. The detail design of T-C-A-C/T

1) Step1 Sampling data: In this step, raw process object data set would be compressed in scale to reduce the difficulty of data mining and a method based on data difference is used to obtain this target. The data section with most maximum amount of change would be chosen as sample data set, for the data difference means the state changing, and the more amount of change, and the data section with the most amount of change would be most appropriate to reflect the state of whole data set. The detail procedure of this step is as follows:

Assuming the process object $\mathcal{X}$ has n links as $\mathcal{X} = \{X_1, X_2 \dots X_n\}$, and $x_i(t_j)$ is a measured value of a random link X_i at time t_j. The raw data of $\mathcal{X}$ in time section $T = \{t_1, t_2, \cdots, t_m\}$ is shown as table 2, and $t_1 < t_2 < \cdots < t_m$.

Assuming the difference of a random link X_i is $\Delta x_i(t_j)$ at time t_j, then the value of $\Delta x_i(t_j)$ can be calculated as eq1:

$$\Delta x_i(t_j) = x_i(t_{j+1}) - x_i(t_j) \quad (1)$$

TABLE II. THE MONITORING DATA OF PROCESS OBJECT

T	X_1	X_2	⋯	X_i	⋯	X_n
t_1	$x_1(t_1)$	$x_2(t_1)$	⋯	$x_i(t_1)$	⋯	$x_n(t_1)$
t_2	$x_1(t_2)$	$x_2(t_2)$	⋯	$x_i(t_2)$	⋯	$x_n(t_2)$
⋮	⋮	⋮	⋮	⋮	⋮	⋮
t_j	$x_1(t_j)$	$x_2(t_j)$	⋯	$x_i(t_j)$	⋯	$x_n(t_j)$
⋮	⋮	⋮	⋮	⋮	⋮	⋮
t_m	$x_1(t_m)$	$x_2(t_m)$	⋯	$x_i(t_m)$	⋯	$x_n(t_m)$

If $\Delta x_i(t_j) > 0$, the change trend of link X_i at time t_j is ascending; otherwise if $\Delta x_i(t_j) < 0$, the change trend of link X_i at time t_j is descending. The time section $T = \{t_1, t_2, \cdots, t_m\}$ is divided into w parts in average, and every parts' length is h, and a random part $T_y = \{t_{u_y}, t_{u_y+1}, \cdots, t_{u_y+h-1}\}$, $1 \le y \le w$, and all links' change amount in T_y is $\Delta\mathcal{X}_y$ which can be calculated by eq2.

$$\Delta\mathcal{X}_y = \sum_{i=1}^{n} \sum_{j=t_{u_y}}^{t_{u_y+h-1}} |\Delta x_i(t_j)| \quad (2)$$

Then in $T = \{t_1, t_2, \cdots, t_m\}$, $\exists T_M \subset T$, $T_M = \{t_{u_M}, t_{u_M+1}, \cdots, t_{u_M+h-1}\}$ makes $\Delta\mathcal{X}_M$ have maximum value, and the data set in T_M would be selected as sampling data.

2) Step 2 Adjusting time series of data: Time series is a data list of some parameters value which is arranged by time order. For the transitivity of data in process industry, the change of one link's state would cause other links' changing. Extreme point is a representative data which can be easy observed, so in this paper, the time interval of different links' extreme point is used to determine and adjust the data set's time series.

To a random link$X_i \in$, $\exists[t_l, t_p] \subset T_M$, and $\exists t_o \in (t_l, t_p)$, and $x_i(t_o) > x_i(t_j)$ or $x_i(t_o) < x_i(t_j), t_j \in [t_l, t_o) \cup (t_o, t_m]$, then $x_i(t_o)$ is the extreme value of X_i in time interval $[t_l, t_p]$. Assuming that there are q extreme points in sample data set, the extreme point's value of process object $\mathcal{X}$ is as table3, and the time of extreme point in link X_i can be expressed as $\{t'_1(X_i), t'_2(X_i) \ldots t'_q(X_i)\}$.

TABLE III. TABLE OF DIFFERENCE

number	X_1	X_2	⋯	X_i	⋯	X_n
1	$x_1(t'_1)$	$x_2(t'_1)$	⋯	$x_i(t'_1)$	⋯	$x_n(t'_1)$
2	$x_1(t'_2)$	$x_2(t'_2)$	⋯	$x_i(t'_2)$	⋯	$x_n(t'_2)$
⋮	⋮	⋮	⋯	⋮	⋯	⋮
q	$x_1(t'_q)$	$x_2(t'_q)$	⋯	$x_i(t'_q)$	⋯	$x_i(t'_q)$

In process object, measured data from all links can consist of a time series, and the time interval of different link is Δt_{ij}. If $\Delta t_{ij} > 0$, the link X_j is coming from link X_i, and the change of X_i leads the change of X_j. Otherwise, if $\Delta t_{ij} < 0$, the link X_i is coming from link X_j, and the change of X_j leads the change of X_i. When calculating time series, a random link X_i would be selected as basic link, then the time interval in a state from to X_i other link X_j can be computed as eq3. In eq3, s means the position of X_i with extreme value, and r means the position of X_j with extreme value.

$$\Delta t_{ij}(sr) = \left\{ t'_s(X_j) - t'_r(X_i) \middle| \min\{|t'_s(X_j) - t'_r(X_i)| \right\} \quad (3)$$
$$r = 1, 2, \ldots, q\}$$

Then the ultimate time interval is computed as eq4.

$$t_{ij} = \left\{ \Delta t_{ij}(sr) \middle| \max\left\{ \text{count}\left(\Delta t_{ij}(sr) \right) \right\} \right\} \quad (4)$$

According to the time interval of all links, the time series $L = \{X'_1, X'_2, \cdots, X'_n\}$ can be obtained, and the time interval of link X'_i to X'_j can be expressed as eq5.

$$\Delta t'_{ij} = t(X'_j) - t(X'_i) \quad (5)$$

Based on time interval, the sample data set can be shifted, and link X_i is as center, in which its data should not be shifted, and the data of link X_j should be shifted as eq6.

$$X_j(t) = X_j(t - \Delta t'_{ij}) \quad (6)$$

After this adjustment, the data set's time series has been unified, and based on this data set, discovery association rules of process object can be correct operated.

3) Step3 Clustering based on K-means: In this section, the sample data set with right time series would be classified by classical K-means algorithm with optimized K value. After clustering, every cluster means a state of links. In process industry, data quality can be classified as: Best, Better, Normal, Poor and Very Poor; data changing trends can be classified as: Increasing, Deceasing, and fixed. In K-means, the number of clusters can be set in advance, so it is fit to classify the process object data set. To find best result of clustering, a procedure of optimizing K value is designed in this step. The definition of silhouette coefficient is proposed based on the degree of cohesion and separation to help optimize the K value [12]. The clustering algorithm includes two parts: first, K-means will be operated, and then the value of K will be judged by silhouette coefficient of result. The detail of clustering algorithm is as follows:

In K-means, the distance of every data will be computed as eq7, then the object function as eq8 will be executed to judge whether end the procedure of clustering or not.

$$\text{EUCLID}(p_1, p_2) = \sqrt{\sum_{i=1}^{t} (p_{1_i} - p_{2_i})^2} \quad (7)$$

$$E = \sum_{i=1}^{k} \sum_{p=C_i} ||p - m_i||^2 \quad (8)$$

In eq7, p_1 and p_2 are two different data, and t means the attributes' number of every data; in eq8, C_i is the temporary result of clustering, p is the data in C_i, and m_i is the average value of C_i. K-means algorithm will operate as: first, the target data set and the value of K will be input; then k random data will be chosen as the centers of mass will be chosen; the distance of all data to centers of mass will be computed as eq7 and data will join to the cluster of center of mass; then all clusters' mass centers will be re-computed; repeat these steps until all clusters' mass centers are not changed.

After clustering, the quality of K will be judged by silhouette coefficient which is defined as eq9.

$$s_k = \frac{1}{n} \sum_{i=1}^{n} \frac{b_i - a_i}{\max(a_i, b_i)} \quad (9)$$

In eq9, a_i is the average distance from data i to other data in same cluster, and the average distance from data i to other clusters without data i will be computed, and the minimum

value is b_i. By eq9, the range of s_k is [-1, 1]. When $s_k > 0$, the result of clustering is reasonable, and the value is more close to 1, the result is better; otherwise, when $s_k < 0$, and the value is more close to -1, the result is worse. The K value with largest $s_k$is the best classifying number of process object.

4) Step4 Obtaining association rules: In this section, the association rules of process object will be obtained by improved Apriori algorithm, in which the input data set has been clustered by optimized K value. By the demanding of process object, the rules should be generated from 2-frequant items set, and the rules just include former item and back item, because multi items rules include uncertain factors in process which hardly obtain useful knowledge. These rules mean the relationship of different links in two clusters which satisfy the minimum support and minimum confidence meanwhile. The illustration of rules is as table 4. In table 4, two random links X_i and X_j $(i, j \in \{1,2 \dots n\}, i \neq j)$, there is an association rule between X_i and X_j as $X_i \; in \; k_{ia} \rightarrow X_j \; in \; k_{jb}$, and k_{ia} and $k_{jb}$are two clusters.

TABLE IV. TABLE OF ASSOCIATION RULES IN CLUSTERS

No	Former items	Back items
1	X_i in k_{ia}	X_j in k_{jb}
2	X_i in k_{jc}	X_j in k_{jd}
3	X_j in k_{ja}	X_i in k_{id}
4	X_j in k_{jc}	X_l in k_{la}
⋮	⋮	⋮
s	X_n in k_{ne}	X_{n-1} in $k_{(n-1)a}$
⋮	⋮	⋮
z	X_n in k_{nf}	X_{n-1} in $k_{(n-1)c}$

To reduce the redundancy of association rules, a new conception named associated degree is defined based on the degrees of support $S(ia \rightarrow jb)$ and interest $I(ia \rightarrow jb)$, and $S(ia \rightarrow jb)$ is defined as eq10 and $I(ia \rightarrow jb)$ is defined as eq11.

$$S(ia \rightarrow jb) = P(ia, jb) = \frac{|ia, jb|}{|T|} \quad (10)$$

$$I(ia \rightarrow jb) = \frac{C(ia \rightarrow jb)}{S(jb)} = \frac{P(jb|ia)}{P(jb)} \quad (11)$$

In eq10, $|\text{ia}, \text{jb}|$ is the number of transaction of X_i in k_{ia} and X_j in k_{jb} exists simultaneous; $|T|$ means the total number of transactions. In eq11, $C(\text{ia} \rightarrow \text{jb})$ is the confidence degree of this rule, and $S(\text{jb})$ is the support degree of back item. If $I(\text{ia} \rightarrow \text{jb}) = 1$, X_i in k_{ia} and X_j in k_{jb} are independence of each other; if $I(\text{ia} \rightarrow \text{jb}) > 1$, X_i in k_{ia} and X_j in k_{jb} are positively related, and the value is larger, the promoting effect of X_i in k_{ia} to X_j in k_{jb} is more obvious; if $I(\text{ia} \rightarrow \text{jb}) < 1$, X_i in k_{ia} and X_j in k_{jb} are negative related, and the value is smaller, the inhibiting effect of X_i in k_{ia} to X_j in k_{jb} is more obvious.

Based on the definitions of eq10 and eq11, the conception of associated degree is defined as eq12.

$$c_{ij}(\beta) = S_\beta(\text{ia} \rightarrow \text{jb}) \times \left|I_\beta(\text{ia} \rightarrow \text{jb})\right|, \beta \leq \omega_{ij} \quad (12)$$

In eq12, β means the sequence number of this rule, ω_{ij} means the total number of rules between X_i in k_{ia} to X_j in k_{jb}. Based on eq12, the eq13 defines the conception of associated degree of all rules from link X_i to link X_j.

$$c_{ij} = \sum_{\beta=1}^{\omega_{ij}} c_{ij}(\beta) = \sum_{\beta=1}^{\omega_{ij}} \left(S_\beta(\text{ia} \rightarrow \text{jb}) \times I'_\beta(\text{ia} \rightarrow \text{jb})\right) \quad (13)$$

If $c_{ij} > 0$, the value is larger, the promoting effect of link X_i to link X_j is more obvious; if $c_{ij} < 0$, the value is larger, the inhibiting effect of link X_i to link X_j is more obvious; and if $c_{ij} = 0$, link X_i and link X_j are independence of each other.

5) Step5 Establishing association rules chain/tree: In last section, the association rules have been obtained, but these rules can only reflect the relation of multiple links. In process object, the relation of multiple links should be as an association chain, and in this chain, there should be no reduplicated point, and an entire chain is a complete process or sub-process. The association chain can be defined: there are d links which are not reduplicated each other in a single-direction chain as $\varphi = \{X_1, X_2, \cdots, X_d\}$, and in φ all links are satisfied with time series of process object. The change in one link of association chain will lead to corresponding change of other links in same chain. Assuming the state of link is s_i: if $s_i = 1$, the state of this link will increase; if $s_i = 0$, the state of this link will stay unchanged; if $s_i = -1$, the state of this link will decrease. The state chain will be created by obtaining state status of all links in association chain. State chain is as $\phi = \{\zeta_1, \zeta_2, \cdots, \zeta_d\}$, and ζ_i is a state status corresponding to link X_i.

Based on the calculation of last step, every link's rules' association degree has been obtained, and choosing the links with largest association degree, and arranging links as the order of rules, a strongest association chain can be established. The detail of strongest association chain is as: choosing a random link X_i as the head node, then finding the rule in which the former item is X_i and has the largest associated degree, then if the back item of this rule is not in chain, it will be added to this chain; then finding the rule in which the former item is the second link with largest associated degree, judging the back item whether in this chain or not to decide to add this link to chain; repeat these operations until this chain can't find any new link. Let each link as head node of chain, all strongest association chains can be established.

If these association chains can't include all links in process object, the association tree will be established. In association tree, every branch is an association chain. In establishing association tree procedure, when an association chain can't find any new link, it will flash back to last link and find the back item with second-largest association degree. Repeat these operations until all links have been in this tree. By association tree, the relationships of all links can be obtained.

6) Step5 Establishing state chain to express knowledge: In this section, the state chain will be established based on

association chain, in which every item is computed by the difference of links state status as eq14..

$$\begin{cases} \zeta_i(t_j) = 1, \ \Delta x_i(t_j) > 0 \\ \zeta_i(t_j) = 0, \ \Delta x_i(t_j) = 0 \\ \zeta_i(t_j) = -1, \ \Delta x_i(t_j) < 0 \end{cases} \quad (14)$$

In eq14, $\zeta_i(t_j)$ means the state of a random link X_i at the time t_j. By eq14, the table of links' state is shown as table 5.

TABLE V. TABLE OF STATES CHANGING IN EACH LINK

T	X_1	X_2	⋯	X_i	⋯	X_n
t_{u_M}	$\zeta_1(t_{u_M})$	$\zeta_2(t_{u_M})$	⋯	$\zeta_i(t_{u_M})$	⋯	$\zeta_n(t_{u_M})$
t_{u_M+1}	$\zeta_1(t_{u_M+1})$	$\zeta_2(t_{u_M+1})$	⋯	$\zeta_i(t_{u_M+1})$	⋯	$\zeta_n(t_{u_M+1})$
⋮	⋮	⋮	⋯	⋮	⋯	⋮
t_{u_M+h-1}	$\zeta_1(t_{u_M+h-1})$	$\zeta_2(t_{u_M+h-1})$	⋯	$\zeta_i(t_{u_M+h-1})$	⋯	$\zeta_n(t_{u_M+h-1})$

According to state chain, the relationship of all links' state can be clearly observed. Based on the analysis of state chain, the directive knowledge for process industry object can be obtained. The specific application of state chain can be summarized as:

Reduce the production cost. By browsing state chain, find the state of input link and the intermediate links when final output link increases. Then according to the affection of input links to output link, producer can adjust the state of input to reserve production cost.

Predicate the optimal state of process object. According to state chain, the state of initial link and intermediate links which leads to output state maximum can be found. Then adjusting the corresponding links can make optimal all links state to ensure resource used reasonable.

Detect fault of production process. If there is one link in fault, it can be discovered which links' state will lead to this fault by the state chain. Then according to the changing trend of corresponding links' state, the fault can be detected in advance.

V. EXPERIMENT AND ANALYSIS

In this section, the real data collected in a typical industry process object of thermal power is imported to the algorithm of this paper, and the result of experiment is analyzed. In the procedure of thermal power, a large number of control data is generated every day, and multiple links in the whole production process has obvious characteristics of process object. For the confidentiality of enterprise demanding, this experiment adopts a part of links of thermal power which is mainly of boiler system. The links of experiment is as table 6, and the sampling interval is 5 seconds. The data set is dealt with in the platform of IBM SPSS Modeler 14.1.

TABLE VI. TABLE OF DESCRIPTION OF LINKS IN EXPERIMENT

Link ID	Link name	Description
X_1	10HNA10CQ101_3S	oxygen levels in A
X_2	10HNA20CQ101_3S	oxygen levels in B
X_3	01_q2	q2 in production
X_4	01_q4	q4 in production
X_5	01_q3	q3 in production
X_6	01_Qnetar	lower calorific value
X_7	01_Vdaf	Volatile
X_8	MSTMFLOW	The boiler output

In this experiment, the raw data set is as figure 3, and after sampling function, the data set collected In 2014 July is chosen as sample data set. Then the difference of sample data is computed as figure 4.

dt	x1	x2	x3	x4	x5	x6	x7	x8	x9
2012-06-10 00:...	616.7454	-2.359863	0.2271988	133.8735	133.8735	128.6361	43.71915	234.7799	70.82313
2012-06-10 00:...	615.9319	-2.359676	0.227155	133.8735	133.8735	128.6616	43.55698	232.5577	70.82313
2012-06-10 00:...	609.9911	-2.359488	0.2271111	133.8735	133.8735	128.691	43.41965	231.6997	70.82313
2012-06-10 00:...	602.1344	-2.359301	0.2270672	133.8735	133.8735	128.7263	43.28232	231.7651	70.82313
2012-06-10 00:...	598.6649	-2.359113	0.2270233	133.7434	133.8312	128.6086	43.14499	231.8305	70.82313
2012-06-10 00:...	593.4467	-2.358926	0.2269795	133.7434	133.8735	128.6086	43.00766	231.5567	70.82313
2012-06-10 00:...	590.6846	-2.358739	0.2269356	133.6132	133.7434	128.6086	42.87033	231.7749	70.82313
2012-06-10 00:...	588.8828	-2.358551	0.2268918	133.6132	133.7434	128.6086	44.34127	233.6472	70.82313
2012-06-10 00:...	584.9972	-2.358364	0.2268479	133.3248	133.6353	128.6086	47.40667	242.9077	70.82313
2012-06-10 00:...	581.3506	-2.358176	0.226804	133.0212	133.395	128.6086	48.49777	246.7707	70.82313
2012-06-10 00:...	577.9004	-2.357989	0.2267601	132.7176	133.2229	128.6086	48.48431	246.7585	70.82313
2012-06-10 00:...	572.3165	-2.357802	0.2267163	132.528	133.0927	128.6086	48.47084	246.8853	70.82313
2012-06-10 00:...	572.5377	-2.357614	0.2266724	132.3443	132.9626	128.6753	47.81799	247.0121	70.82313
2012-06-10 00:...	570.5896	-2.357427	0.2266285	132.1821	132.8325	128.6086	47.10187	243.7713	70.82313
2012-06-10 00:...	570.0223	-2.357239	0.2265847	132.052	132.7024	128.6086	46.42258	242.7841	70.90617
2012-06-10 00:...	566.629	-2.357052	0.2265408	132.052	132.5723	128.6086	45.73593	240.7368	70.95087
2012-06-10 00:...	564.8683	-2.356865	0.2264969	131.922	132.4422	128.6086	45.10187	237.7043	70.95087
2012-06-10 00:...	562.7649	-2.356677	0.2264531	131.7919	132.3512	128.4909	44.86594	235.5011	70.95087
2012-06-10 00:...	558.7616	-2.35649	0.2264092	131.6618	132.3122	128.4909	44.63002	237.5619	70.82313
2012-06-10 00:...	559.0934	-2.356302	0.2263653	131.5318	132.1821	128.3733	43.97824	234.8778	70.82313
2012-06-10 00:...	560.2495	-2.356115	0.2263215	131.5318	132.052	128.3733	43.98587	234.7046	70.82313

Fig. 2. The raw data set of thermal power

dt	d_x1	d_x2	d_x3	d_x4	d_x5	d_x6	d_x7	d_x8
2014-07-22 00:00:20.000	-0.0572150000	-0.0209060000	-0.0004100000	-0.0000071000	-0.0000005639	0.0000000000	0.0000000000	-1.0760000000
2014-07-22 00:00:25.000	-0.0572140000	-0.0209050000	-0.0004100000	-0.0000072000	-0.0000005640	0.0000000000	0.0000000000	-0.6730000000
2014-07-22 00:00:30.000	-0.0572150000	-0.0209060000	-0.0004100000	-0.0000071000	-0.0000005639	0.0000000000	0.0000000000	-0.9000000000
2014-07-22 00:00:35.000	-0.0572150000	-0.0209060000	-0.0004100000	-0.0000072000	-0.0000005640	0.0000000000	0.0000000000	-1.0510000000
2014-07-22 00:00:40.000	-0.0572150000	-0.0209060000	-0.0004100000	-0.0000071000	-0.0000005640	0.0000000000	0.0000000000	-1.1270000000
2014-07-22 00:00:45.000	-0.0572150000	-0.0209060000	-0.0004100000	-0.0000072000	-0.0000005640	0.0000000000	0.0000000000	-1.1770000000
2014-07-22 00:00:50.000	-0.0572150000	-0.0209070000	-0.0004100000	-0.0000071000	-0.0000005639	0.0000000000	0.0000000000	-0.5210000000
2014-07-22 00:00:55.000	-0.0572150000	-0.0209060000	-0.0004100000	-0.0000072000	-0.0000005640	0.0000000000	0.0000000000	-0.0850000000
2014-07-22 00:01:00.000	-0.0572150000	-0.0209060000	-0.0004100000	-0.0000071000	-0.0000005639	0.0000000000	0.0000000000	-2.4040000000
2014-07-22 00:01:05.000	-0.0291550000	-0.0209060000	-0.0026410000	-0.0000071000	-0.0000005640	0.0000000000	0.0000000000	-3.9520000000
2014-07-22 00:01:10.000	0.0129350000	-0.0209050000	-0.0026420000	-0.0000072000	-0.0000005640	0.0000000000	0.0000000000	-1.7320000000
2014-07-22 00:01:15.000	0.0129340000	-0.0209060000	-0.0026410000	-0.0000071000	-0.0000005640	0.0000000000	0.0000000000	-0.2520000000
2014-07-22 00:01:20.000	0.0129350000	-0.0209060000	-0.0026410000	-0.0000072000	-0.0000005639	0.0000000000	0.0000000000	-2.2190000000
2014-07-22 00:01:25.000	0.0129350000	-0.0209060000	-0.0026420000	-0.0000071000	-0.0000005640	0.0000000000	0.0000000000	-3.5300000000
2014-07-22 00:01:30.000	0.0129350000	-0.0209060000	-0.0026410000	-0.0000072000	-0.0000005639	0.0000000000	0.0000000000	-3.2520000000
2014-07-22 00:01:35.000	0.0129340000	-0.0209060000	-0.0026410000	-0.0000071000	-0.0000005640	0.0000000000	0.0000000000	-3.0660000000
2014-07-22 00:01:40.000	0.0129350000	-0.0209060000	-0.0026420000	-0.0000071000	-0.0000005640	0.0000000000	0.0000000000	-2.5860000000
2014-07-22 00:01:45.000	0.0129350000	-0.0209060000	-0.0026410000	-0.0000072000	-0.0000005639	0.0000000000	0.0000000000	-2.2670000000
2014-07-22 00:01:50.000	0.0129350000	-0.0209060000	-0.0026410000	-0.0000071000	-0.0000005640	0.0000000000	0.0000000000	-2.0390000000
2014-07-22 00:01:55.000	0.0245190000	-0.0061870000	-0.0026420000	-0.0000072000	-0.0000005640	0.0000000000	0.0000000000	-1.8880000000

Fig. 3. The difference of sample data

In figure 4, "d_xi" means the first order difference of link X_i. It can be seemed that in the time interval of this figure, the trend of X_1 is increasing at first and then decreasing; the trends of X_2, X_3, X_4 and X_5 are increasing steady; the trend of X_8 is changed large; and the differences of X_6 and X_7 are zero, it means these links are not changed.

Based on the result of differences, the extreme points of all links can be obtained. In figure 5, the extreme points of link is X_2 is shown.

dt	fldname	extremum
2014-07-23 10:18:10.000	d_x2	-0.02457
2014-07-23 10:21:10.000	d_x2	-0.00561
2014-07-23 10:30:30.000	d_x2	0.01836
2014-07-23 10:35:25.000	d_x2	0.00506
2014-07-23 10:36:55.000	d_x2	-0.00080
2014-07-23 10:39:25.000	d_x2	0.00155
2014-07-23 10:49:30.000	d_x2	-0.01487
2014-07-23 10:51:30.000	d_x2	0.00623
2014-07-23 10:52:15.000	d_x2	-0.01180
2014-07-23 10:55:35.000	d_x2	-0.00269
2014-07-23 10:56:00.000	d_x2	0.01700
2014-07-23 10:56:15.000	d_x2	-0.04626
2014-07-23 10:56:30.000	d_x2	0.05775

Fig. 4. The extreme points of link X_2

In figure 5, "dt" is the time of extreme point appearing of links, and "fldname" is the link's name, and "extremum" is the

value of difference. After obtaining all links' data of extreme points, the time interval of every pair of links should be calculated. In this paper, the link X_1 is acting as reference point, and the time intervals from X_1 to other links are shown in figure 6.

	fldname	adjust
1	x1	0
2	x2	0
3	x3	45
4	x4	565
5	x5	3380
6	x6	0
7	x7	5
8	x8	0

Fig. 5. The Time intervals from X_1 to other links

In figure 6, "adjust" is the time interval, and the values in X_2, X_6 and X_8 are zero, so these links' time interval is less than 5s, and these links can be seemed operate synchronously; the values in X_7, X_3 and X_4 increase progressively, and these links can be seemed operated in order behind X_1, X_2, X_6 and X_8. Then these 8 links' time series are as eq15. Based on these time series, the raw sampling data set and their difference should be adjusted by "adjust" in figure 6.

$$L = \{\{X_1, X_2, X_6, X_8\}, \{X_7\}, \{X_3\}, \{X_4\}, \{X_5\}\} \quad (15)$$

When the sampling data set has been adjusted the time series, it can be clustered by improved K-means. At first, the optimal K value of every link can be computed, and the values of links X_2, X_6 and X_8 are larger than else, which means these links' state change obviously. Then based on K value, the clustering result is computed.

Based on the result of clustering, 2-items association rules can be discovered as figure 7, and in figure 7, "preitem" is the former item of rule, and "nextitem" is the back item of rule, and "lift" is the interest degree of this rule. In these rules, only the rule whose lift is larger than one will be kept.

	preitem	nextitem	support	confidence	lift
	x1=clusters-1	x4=clusters-2	0.5498702583...	0.5047011673...	0.9919622575...
	x1=clusters-1	x5=clusters-1	0.5498702583...	0.6827836614...	0.8649293797...
	x1=clusters-1	x6=clusters-1	0.5498702583...	0.5660718135...	0.9575296087...
	x1=clusters-1	x7=clusters-2	0.5498702583...	0.5558411092...	1.0088874915...
	x1=clusters-2	x4=clusters-2	0.4501297416...	0.5137863892...	1.0098187591...
	x1=clusters-2	x5=clusters-1	0.4501297416...	0.9196621676...	1.1649997988...
	x1=clusters-2	x6=clusters-1	0.4501297416...	0.6218504471...	1.0518810531...
	x1=clusters-2	x7=clusters-2	0.4501297416...	0.5449631001...	0.9891432029...
	x1=clusters-2	x8=clusters-4	0.4501297416...	0.7306070904...	2.1109864277...
	x2=clusters-1	x1=clusters-2	0.1640783676...	0.5189746682...	1.1529446295...
	x2=clusters-1	x4=clusters-2	0.1640783676...	0.5407794660...	1.0628721601...
	x2=clusters-1	x5=clusters-1	0.1640783676...	0.8554102408...	1.0836074306...
	x2=clusters-1	x6=clusters-1	0.1640783676...	0.6400075090...	1.0825943371...
	x2=clusters-1	x7=clusters-2	0.1640783676...	0.5474124935...	0.9935890100...

Fig. 6. Pars of association rules

Based on the result of figure 7, the strongest association chain can be created, all links' strongest association chain are shown as table 7.

TABLE VII. TABLE OF ALL LINKS' STRONGEST ASSOCIATION CHAIN

Links' ID	The strongest association chain
X_1	$\varphi_1 = \{X_1, X_8, X_5\}$
X_2	$\varphi_2 = \{X_2, X_1, X_8, X_5\}$
X_3	$\varphi_3 = \{X_3, X_4, X_5\}$
X_4	$\varphi_4 = \{X_4, X_5\}$
X_5	—
X_6	$\varphi_6 = \{X_6, X_4, X_5\}$
X_7	$\varphi_7 = \{X_7, X_4, X_5\}$
X_8	$\varphi_8 = \{X_8, X_1, X_7, X_4, X_5\}$

By this table, it can be seemed that all strongest association chains are satisfied with the time series L as eq15. And the association tree can be established as figure 8. In this tree, all strongest association chains are included, and every branch of the tree is one strongest association chain, and more information is included in this tree. The tree of link X_1 is shown in figure 8, and the number in every branch is the value of association degree of former item to back item.

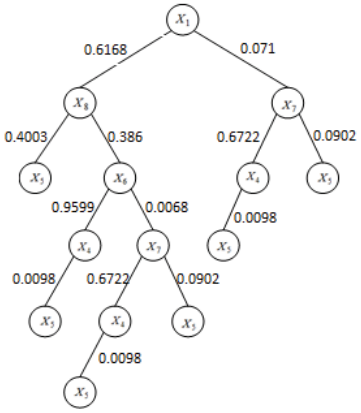


Fig. 7. The association tree of link X_1

Through association chain/tree, the state chain can be established, and the detail data of all state chains of association chain φ_7 is listed in table 8. In table 8, the first four chains account for the largest proportion, and the number is 75.893%, so these state chains can be seemed as normal states. The 7th association state chain has lowest proportion of all, and its state contradicts other chains obviously, so it can be seemed as fault state. The 5th and the 6th association state chains' proportion is lower than the No.1-4, but it is larger than the 7th one, and the states in these chains don't contradict normal chains, so these states can be seemed transient states.

TABLE VIII. TABLE OF THE STATE CHAINS OF φ_7

No	X_7	X_4	X_5	Percentage (%)	Description
1	Unchanged	Decrease	Decrease	20.875	Normal
2	Unchanged	Increase	Increase	20.467	Normal
3	Unchanged	decrease	Increase	17.442	Normal
4	Unchanged	Increase	Decrease	17.109	Normal
5	Unchanged	Unchanged	Decrease	11.17	Transient
6	Unchanged	Unchanged	Increase	7.392	Transient
7	Increase	Increase	Decrease	1.271	Fault

Based on these state chains, the advices can be concluded as: if the output of link X_5 wants to be improved, the count of link X_4 should be added with the unchanged count of link X_7; by browsing these chains, finding the chain that X_4 and X_5 increase at the same time, and the value of X_4 that makes the output of X_5 is maximum can be discovered, and this value is the best value; if there is some faults in X_5, the value of X_4 and X_7 should increase by these states, so monitoring the states of X_4 and X_7 can defect the fault of X_5 in advance.

These advice generated by real data, and the effectiveness has been proved by experts of corresponding production field, and it can be concluded that our researches of process object not only can discover the inner relationship of each link in production, but also can guide to improve the process of production.

VI. CONCLUSION

In this paper, the characteristics of process object is analyzed, and based on existing researches, the algorithm named T-C-A-C/T is proposed in this paper. In this algorithm, there are several innovations: the large scale of history data of process industry is reasonable sampling which reduce the difficulty of dealing with; the process object's time series can be discovered and the data set is also adjusted by time series which ensure the knowledge is fit for process demand; and the association rules are discovered by improved Apriori algorithm; and the knowledge is expressed by the means of states chain which is more clearly to reflect the relationship of process object. At last, the performance of T-C-A-C/T is proved by the experiment with real data of thermal power.

ACKNOWLEDGMENT *(Heading 5)*

This research is supported by Natural Science Foundation of China under the Contract Number 60573065; and is supported by Natural Science Foundation of Shandong under the Contract Number ZR2012FL12; and is supported by The Science and Technology Development Plan of Shandong under the Contract Number 2014GGX101039.

REFERENCES

[1] Srikant R， Agrawal R. Mining Quantitative Association Rules in Large Relational Tables[C]. Proceedings of 1995 Very Large Databases Conference

[2] Cheyng D W， Han K， Ng V， et al. Maintenance of Discovered Association Rules in Large Databases： An incremental updating technique[C]. Proceedings of the 1996 International Conference on Data Engineering.

[3] L.X. Wang， J.M. Mendel， Generating fuzzy rules by learning from examples[J]， IEEE Transactions on Systems， Man and Cybernetics 22 (1992) 1414–1427.

[4] QiSen Cai, DeF Zhang, Bo Wu, Stehpen C.H.Leung. A novel stock forecasting model based on fuzzy time series and genetic algorithm[J]. International Conference on Computational Science, 2013.

[5] M. Gas, K. lin, H. Mannila, G. Renganathan. Rule Discovery From Time Series[C]. Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, 1998.

[6] Marina Resta. Seize the (intra)day: Features selection and rules extraction for tradings on high-frequency data[J]. Neurocomputing, 2009.

[7] Imran Khan, Arun Kulkarni. Knowledge Extraction from Survey Data Using Neural Networks[J]. Procedia Computer Science, 2013.

[8] Erol Egrioglu. PSO-based high order time invariant fuzzy time series method: Application to stock exchange data[J]. Economic Modelling, 2014.

[9] R. Agrawal, C. Faloutsos, and A. Swami. Efficient similarity search in sequence databases. In Proc. of the Fourth International Conference on Foundations of Data Organization and Algorithms, Chicago, October 1993.

[10] J. Han, M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann: 2000

[11] Zaki MJ, Hsiao CJ. CHARM: An efficient algorithm for closed association rule mining. Tehnical Report, TR99-10, Department of Computer Science, Rensselaer Polytechnic Institute,1999.

[12] Pang-Ning Tan, Michael Steinbach, Vipin Kumar. Introduction to Data Mining[M]. Pearson Education, 2006: 330-336

SESSION

POSTER PAPERS

Chair(s)

TBA

Circadian rhythm of hospital death: difference between the intensive care unit and general room

Jung-Ho Park[1], Yun-Kyeong Cho[1], Jong-Ha Lee[2], Yunjung Lee[3], In-Cheol Kim[1], Hyoung-Seob Park[1], Hyuck-Jun Yoon[1], Hyungseop Kim[1], Chang-Wook Nam[1], Seongwook Han[1], Seung-Ho Hur[1], and Yoon-Nyun Kim[1]

[1]Department of Internal Medicine, College of Medicine, Keimyung University, Daegu, Korea

[2]Medical Imaging and Biosignals Laboratory, Department of Biomedical Engineering, College of Medicine, Keimyung University

[3] R&BD Center for Medical Devices for Arrhythmia Diagnosis and Therapy, Keimyung University Dongsan Medical Center, Daegu, Korea

Abstract - *The purpose of the present study was to record the time at which biological phenomena stop in different hospital wards and determine regular patterns in times of death, as well as any factors involved. A comparative analysis was conducted for the morality distribution between the intensive care unit (ICU) and general rooms (GR), according to year, month, day of week, and hour of day, disease entity, as well as the age and sex of the patients. We collected the data on all deaths in ICU and GR during 7 years and analyzed by cross analysis process, which shows absolute numbers and proportions, for comparison of each wards. There are obvious differences in circadian rhythm between ICU and GR, it may be accounted for by the presence of preserved circadian rhythm.*

Keywords: Circadian rhythm, Chronobiology phenomena, Biological clocks

1 Introduction

Circadian rhythms are vital in adapting to changes in one's environment and sustaining life. Existing research illuminates the effects of changes in the circadian rhythms of animals. The suprachiasmatic nucleus (SCN) of the hypothalamus plays the most important role in the maintenance of the mammalian biological clock [1,2]. The SCN receives photosensitive information through the eyes, and its malfunction completely eliminates rhythms concerning regular sleeping and waking patterns. The retina includes not only regular photoreceptors, which participate in visual perception, but also light-sensitive photosensitive ganglion cells. These cells contain photopigments called melanopsin, through which they detect light. The detected light travels via the retinohypothalamic tract to the SCN, thus affecting biological rhythms. That is, the SCN receives information regarding the length of day and night via the retina and interprets it, after which it signals the pineal gland of the hypothalamus to secrete melatonin. The melatonin secretion level is at its highest during the night and lowest during the day. However, centuries of civilization and industrialization, and changes in living patterns and hobbies in humans are speculated to have brought about a change of behavior in daily, weekly, and even yearly activities of people, and thus caused human circadian rhythms to evolve. Death occurs when biological regulatory functions that maintain homeostasis stop working. Existing studies suggest that circadian rhythms and biological clocks may affect homeostasis, although results of previous research studies do not show consensus on the matter. The purpose of the present study was to record the time at which biological phenomena stop in different hospital wards and determine regular patterns in times of death, as well as any factors involved.

2 Materials and Methods

This study is a retrospective study which enrolled inpatients at the Keimyung University Dongsan Medical Center who died over 7 years, between January 2006 and December 2012. A comparative analysis was conducted for the morality distribution between the intensive care unit (ICU) and general rooms (GR), according to year, month, day of week, and hour of day, disease entity, as well as the age and sex of the patients. We collected the data on all deaths in ICU and GR during 7 years and analyzed by cross analysis process, which shows absolute numbers and proportions, for comparison of each wards. The Statistical significance level was set at 5% and all calculations were two-tailed. The flow of mortality case in ICU and GR was analyzed by annually, monthly, and weekly. The daily pattern of death was divided by 2-hour periods, proportion of death in age was divided by decade. The mortality case in two wards was compared by sex, surgical status and the format of mortality distribution was used by international classification of disease (ICD) categories.

3 Results and Discussion

A total of 6517 patients died, among whom 3198 (49%) were from the Intensive Care Unit (ICU) and 3319 (51%) were from the General Room (GR). Year-wise, of the 3198 (49%) who died in the ICU, 433 (13.7%) died in 2010, 468 (14.6%) died in 2011, and 489 (15.3%) died in 2012, thus

showing an increase over the most recent 3 years. Of the 3319 who died in the GR, 495 (14.9%) died in 2010, 470 (14.2%) died in 2011, and 447 (13.5%) died in 2012, thus showing a decrease over the most recent 3 years (P = 0.06). The number of deaths in the ICU notably increased on a yearly basis despite advances in medical techniques and information.

In terms of months, the ICU saw the highest number of mortalities in January (9.5%) and September (8.8%), whereas the GR saw the highest number of mortalities in January (9.1%) and August (9.0%) (P = 0.87). When analyzed by days of the week, the highest mortality rate occurred in the ICU on Wednesdays (14.9%), as opposed to Sundays (14.8%) in the GR. (P = 0.60).

When mortality rate was examined in 1-hour intervals, no statistical difference was found between the two wards (P = 0.09). When examined in 2-hour intervals, the ICU peaked in mortality between 14–16 (9.2%) and 20–22 hours (9.1%), whereas the GR peaked in mortality between 6–8 (9.6%) and 10–12 hours (9.4%), with a significant statistical difference between the two wards (P = 0.03; Table 1).

Table 1. Mortality rate by hour of day (P = 0.03).

Ward / Hour	ICU	GR	Total
0–2	277 (8.7%)	245 (7.4%)	522
2–4	256 (8.0%)	270 (8.1%)	526
4–6	230 (7.2%)	303 (9.1%)	533
6–8	266 (8.3%)	318 (9.6%)	584
8–10	252 (7.9%)	279 (8.4%)	531
10–12	287 (9.0%)	312 (9.4%)	599
12–14	269 (8.4%)	286 (8.6%)	555
14–16	294 (9.2%)	260 (7.8%)	554
16–18	265 (8.3%)	270 (8.1%)	535
18–20	247 (7.7%)	254 (7.6%)	501
20–22	290 (9.1%)	263 (7.9%)	553
22–24	265 (8.3%)	259 (7.8%)	524
Total	3198 (100%)	3319 (100%)	6517

Differences in mortality rates, as according to time of day or day of the week, have been researched most commonly in terms of cardiovascular diseases [3-7]. In the present study, the high mortality rate in the early morning hours (4–6 AM) in the GR may be corroborated by such studies as shown in Table 1. The reason for the lack of such an effect in the ICU may be that endogenous circadian rhythms are changed as the excretion of melatonin from the pineal gland is affected by the use of various sympathomimetic and sedative drugs, as well as the environment such as a hospital ward, in which daytime and nighttime differences are unclear. Hourly melatonin secretion levels have not been measured in the present study, but future studies may be able to analyze the mortality rate differences between the ICU and GR by investigating the causal relationship between melatonin secretion and circadian rhythms.

When analyzed by age, mortality occurred most often in those aged in their 60s (ICU, 25.7%; GR, 26.2%) and 70s (ICU, 28.5%; GR, 24.2%) (P < 0.01). When analyzed by sex, mortality occurred more so in men in both groups (ICU, 59.8%; GR, 62.0%), but this effect was not statistically significant (P = 0.07). When analyzed by group, the patients whose diseases fell under group I (diseases of the circulatory system) of the ICD died most often in the ICU (28.3%), whereas those whose diseases fell under group C (neoplasms) had the highest mortality rate in the GR (77.7%). The difference was statistically significant (P < 0.01). When analyzed by surgical status, in both wards, the patients who did not receive any surgical treatment had significantly higher mortality than those who did (P < 0.01).

4 Conclusions

The difference between ICU and GR is notable in diurnal variation. The point of this difference is explained by presence of preserved circadian rhythm. Because ICU patients might have been influenced by more factors of environmental changes than GR patients such as differences in disease distribution, hospital room environments, use of various drugs, ICU patients represent changes in circadian rhythm compared with GR patients. However, the limitation of our study is that we have not investigated direct causation of circadian variation and such factors mentioned above.

5 References

[1] Zhou XJ, Jiang XH, Yu GD, Yin QZ. "Modulation of circadian rhythm of discharges of suprachiasmatic nucleus neurons in rat hypothalamic slices by melatonin"; Acta Physiologica Sinica 52(3), 215-219, Jun 2000.

[2] Gillette MU, McArthur AJ. "Circadian actions of melatonin at the suprachiasmatic nucleus"; Behav Brain Res 73(1-2), 135-139,1995.

[3] Panza JA, Epstein SE, Quyyumi AA. "Circadian variation in vascular tone and its relation to alpha-sympathetic vasoconstrictor activity"; N Engl J Med, 325(14), 986-990, Oct 1991.

[4] Muller JE, Ludmer PL, Willich SN, Tofler GH, Aylmer G, Klangos I, et al. "Circadian variation in the frequency of sudden cardiac death"; Circulation 75(1), 131-138, Jan 1987.

[5] Corbalan R, Verrier R, Lown B. "Psychological stress and ventricular arrhythmias during myocardial infarction in the conscious dog"; Am J Cardiol 34(6), 692-696, Nov 1974.

[6] Friedman M. The pathogenesis of coronary plaques, thromboses, and hemorrhages: an evaluative review"; Circulation , 52(6 Suppl), III34-40, Dec 1975.

[7] Mundigler G, Delle-Karth G, Koreny M, Zehetgruber M, Steindl-Munda P, Marktl W, et al. "Impaired circadian rhythm of melatonin secretion in sedated critically ill patients with severe sepsis"; Crit Care Med, 30(3), 536-540, 2002.

Predicting Sports Outcomes with a Rank-And-Choose Variable Selection Process

Tao Lin, Ricardo Trujillo, Haibo Wang
R. Sanchez, Jr. School of Business, Texas A&M International University, Laredo, Texas, USA

Abstract - *Predicting outcomes of games or championship is of great interest to fans, sponsors and media. There are many factors in determining the outcomes of each game or championship. We develop a logistics regression model with a rank-and-choose variable selection process in this study. We implemented this model in open source R language and computed the predictive variable efficiencies from historical data, then test the predictive model for 2015 NBA championship series data with the comparison of Microsoft Bing search engine prediction. The approach we were using for solving this problems includes: 1. Gathering the historical performance data of all teams from the official stats website; 2. Using LASSO method to select independent variables that are strongly correlative to the dependent variables; 3. Building a regression model with R with the historical data, then plug in the data of this season to accomplish the prediction.*

Keywords: Prediction; Sports games; Variable selection; Liner regression; LASSO; R

1 Research Design And Methodology

1.1 Data collection:

We collected the historical team performance data from the official statistics website: NBA.com/stats. We mainly focus on the historical performance data of the champions of each season.

1.2 Research tools:

We use a popular statistics analysis tool—R to do the analysis work. With R, we use the data we collected to select variables, build regression model, test accuracy then plug the data to predict.

1.3 Model Evaluation:

We used LASSO method for selecting variables, by adjusting the model, overall error rate is about 4.89%, and got the same result with Microsoft Bing search engine.

2 System Modeling And Results

2.1 Data collection:

Our data is collected from the official website of NBA (http://stats.nba.com/). As mentioned above, we collect the history performance data including the number of winning games and some remarkable technique indicators. As the independent variables for prediction, we also collected the performance data of the playoff teams in season 2014-2015.

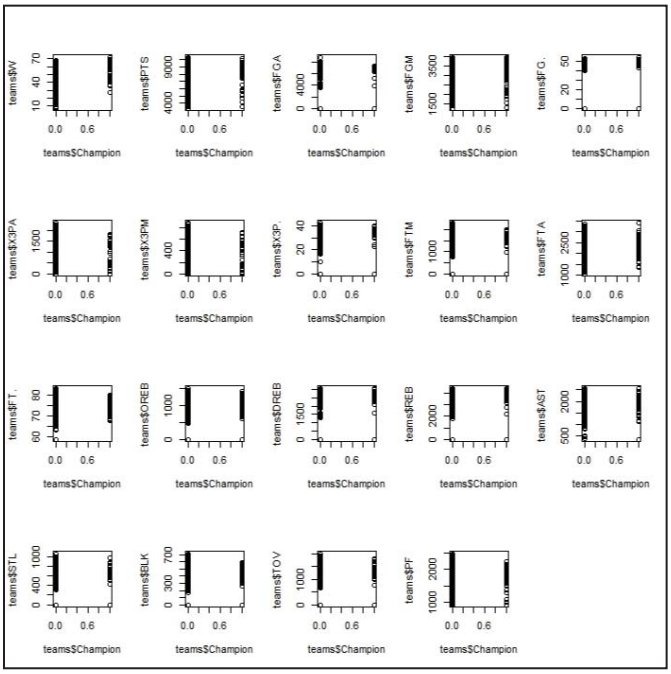
2.2 Data understanding:

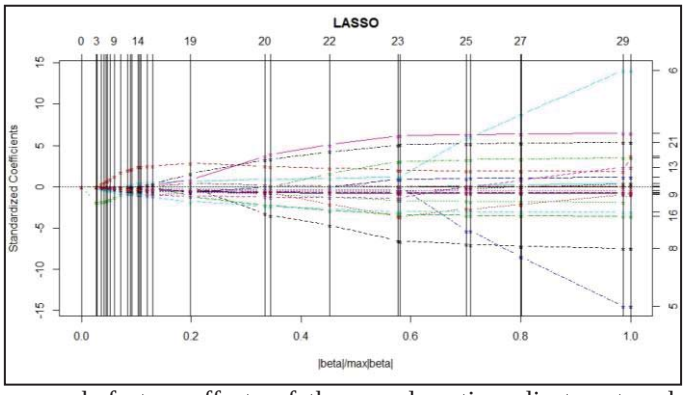
For understanding the relationship of the independent variables, we used multiple plot to find out the correlations between each technique indicators and champion winner.

2.3 Model analyzing:

LASSO is a good tool to find out the relationship among variables and a good combination which can provide good results. In this case, the graph shows there are several key variables and made significant contribution to the model.

To optimize our model, we made 3 cases with different number of variables. At the same time, we made a function to





research factors effects of the sample ratio, adjustment and sampling randomness.

We also divide the data set into training group and test group by the sampling ratio, then use the training data to build a Generalized Linear Model.

2.4 Model testing:

By adjusting the model and trying the factors, the most optimized model will includes all key variables with the specific set of model factors.

2.5 Building Model and Prediction

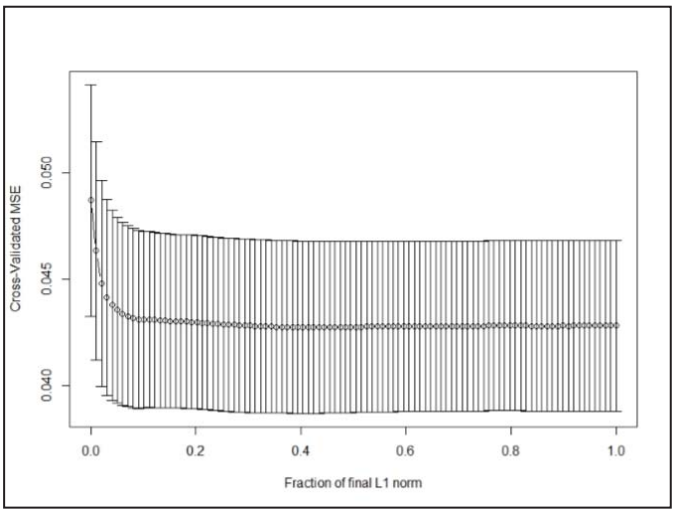
We used the 14-15 seasons play-off teams technique indicators to plug in the model, the result shows the Golden State Worriers is most possibly to be the champion of season 2014-2015.

3 Conclusions

As the conclusion, we used the all the key variables such as number of winning, total points, field shoot attempt, 3 points shoot accuracy, offence and defense rebound, steal and block.

We put the sampling rate and adjustment to 80%, also we use seed 4 as the sampling randomness. In this condition, the error rate will be 4.89% which is the optimized condition.

We also gathered the performance data of the playoff teams and plug into the model, and we have the prediction that the Golden state worriers is most likely to be the champion of season 2014-2015.



4 Future Work

We will focus on improve our model's accuracy by reevaluating other technique indicators, try to find more efficient indicators.

We will also try to use this method to do more researches on other popular sports event.

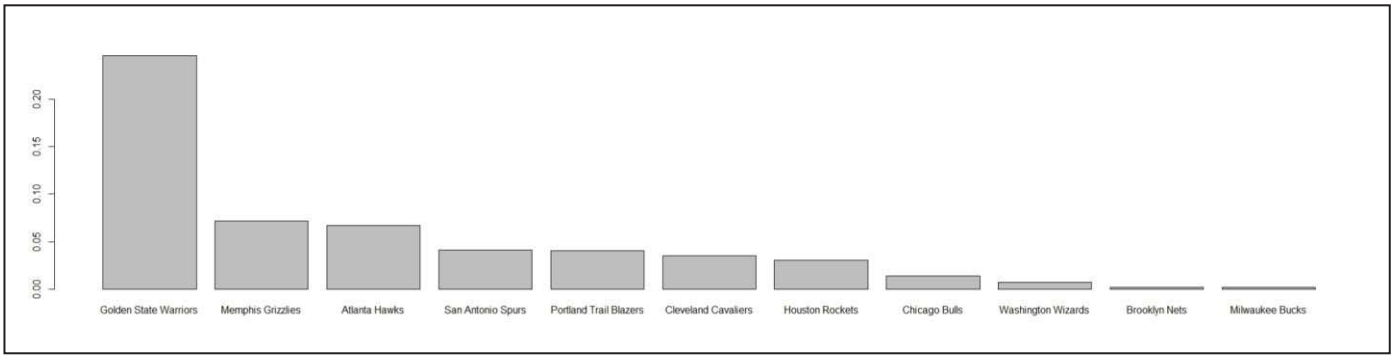
5 References

[1] Kabacoff, R. , (2011). R in action: Data analysis and graphics with R. [Online]. [May. 10, 2015].

[2] Ledolter, Johannes. , Data Mining and Business Analytics with R..

[3] Wikimedia Foundation,Linear Regression [Online]. Available: https://en.wikipedia.org/wiki/Linear_regression

[4] NBA.com,NBA.com/Stats [Online]. Available: NBA.com/Stats.



The Impact of ICT and Big Data on e-Government

Joseph M. Woodside[1], **Shahram Amiri**[2], **and Brianne Boldrin**[2]
[123]Department of Decision and Information Sciences, Stetson University, DeLand, FL, USA

Abstract - *Given the global debt crisis, increasing government debt ratios are unsustainable. An e-government model utilizing ICT and resulting Big Data is anticipated to reduce costs and decrease debt. This poster paper develops the research background methodology and outline for development of an e-Government model using global indicators.*

Keywords: Big Data, e-Government, ICT

1 E-Government, ICT and Big Data

Current and developing technologies promote the advancement of E-government. E-government uses information and communication technologies (ICT) to allow the government to connect with citizens and private groups electronically. This provides citizen centered services that increase the transparency of governmental agencies through the integration of various departments and programs. Two major trends have increased E-governance projects for ICT. The first is the recent development in the ease of the use of the governmental operational systems. The second is the increasing use of ICT in the daily lives of citizens, growing the community's level of knowledge and skill [1]. Literature says, "E-government systems frequently encompass strategic goals that go beyond efficiency, effectiveness and economy to include political and social objectives, such as trust in government, social inclusion, community regeneration, community wellbeing and sustainability" [2].

Big data, cloud computing technology and increased departmental communication have been described as recent development trends in E-Government. Big data is the massive amount of digital data that is collected and compiled from a variety of different sources. Sources say, "90% of the world's data today was generated during the past two years, with 2.5 quintillion bytes of data added each day". A large amount of this data is unusable to relational databases because of its structure. Resources are need to transform this data into more than just figures. Data is a valuable resource that has new value and can increase discovery and business intelligence. Governments can use big data to help serve their citizens and overcome national challenges such as rising health care costs, unemployment, natural disasters and terrorism [3]. Recently, the Obama admiration began a Big Data Research and Development Initiative with the intent to "improve [American] capability to extract knowledge and insights from large and complex collections of digital data; harness these technologies to accelerate the pace of discovery in science and engineering; strengthen national security and transform teaching and learning" [4]. Big data will help enhance the use of E-government along with the use of several other technologies.

2 Cost Savings from ICT and Big Data

Today's literate provides a limited amount of data and research that significantly supports the overall cost savings of the government with the implementation of an ICT system [5]. However, many works of literature do note the cost savings that the ICT provides to citizens once the system is implemented. Recently, a study was conducted with the Road Safety and Transport Authority in Bhutan. The study analyzed the transition of several offices from a manual system of issuing drivers licenses and automobile licenses to an electronic database system. The old system required that the licenses be sent from four regional offices to a main office for approval. This resulted in a large amount of backorders and a long duration of travel time (at least 2 weeks each). With automobile documents requiring renewal every year and drivers licenses every five years, the volume of documents was very high. The government hoped that an electronic system could help streamline this process and reduce turnover time so a new IS system was installed in November of 2004. The system was completed less than a year later in July of 2005. In order to retrieve substantial data, the licensing system was studied before the IS implementation as well after the new system had been fully functioning for a significant amount of time, in 2007 [6].

Data showed that the activity based cost of direct labor fell 24%. However, the costs to implement the system itself rose 43% [6]. Literature cites a high rate of project failure, the learning curve and the need to implement a system simultaneously to an existing system, as some of the challenges of ICT similar to the project in Bhutan [7]. However, there are several cost and service benefits the citizens receive through successful implementation. The customers have better service, a higher level of quality and fairer treatment. This can allow the customer to cut down on travel expenses and the time that they spend away from work or waiting to get a license. A similar study conducted at seven service locations in rural and urban India also concluded comparable findings. They found that after the various projects began using ICT systems, the number of trips citizens made to complete transactions for a service were reduced at

all seven locations, thus reducing the citizens travel costs. They also found that the wait time at all seven locations was reduced 30-60%, resulting in a reduction in forgone wages [5].

Recent publications also recognize cost savings through the correct research and use of big data. The British government could make a savings of £33bn through better use of big data [8]. This large predicted figure is the result of better data collection, storage, analysis and interpretation. Government officials stated that the use of data should increase the quality of decision-making and increase the collaboration between departments. Currently, the siloes in government lead to bad decision making because of the poor and restricted sharing of data and communication. With an improved big data system, information can become more accessible from all departments and can be exchanged more frequently. Big data can also help the government by decreasing the amount of fraud that occurs after the ability to complete a more comprehensive analysis on tax submissions data. Data can also be used to be the base of a health care system that is more focused on preventative care the wellbeing then chronic care. This will create a much greater return for the public information [8].

3 The Global Debt Crisis

It may appear that a countries capability to produce hinges on the workers and machines that it employs despite its balance sheet, however, mass production doesn't always result in overall wealth [9]. While production and economic growth have increased across the world, the combined debt that counties owe, including governments, corporations, banks and households, has also increased dramatically. Literature claims that since the global crisis at the end of 2007, world debt has increased a whopping 57 trillion dollars or 286% of global economic output. Since 2007, China, one of the largest growth economics in the world, recorded a ratio of debt to economic output up 83 percentage points. This debt can have catastrophic worldwide impacts. It has been cited that public or private high debt, causes economy's to be increasingly vulnerable to shifts in the economy and can act as the fuel to extreme booms or busts [9].

In the early 2000's, five countries were predicted to act as the future "drivers of global growth". These five countries were: Brazil, Russia, India, China and South Africa. The acronym BRICS was coined to represent these five countries of promising markets for financial capital. While each of these countries are unique, literature offers support for some similarities that may have helped these countries produce some of the largest economic growth in the world. One of these similarities is the marriage between global capital and cheap labor. China and Russia both have capitalist regimes with the capability to control workers [10]. Slovenian philosopher Slavoj Zizek writes that China "seems to embody a new kind of capitalism: disregard for ecological consequences, disdain for workers' rights, everything subordinated to the ruthless drive to develop and become the new world force" [11]. Conversely, Brazil, South Africa and India have electoral democracies, however; these countries also have power central bureaucracies that choose profits over welfare. These governmental characteristics have caused dramatic implications within each of the BRICS. Many social, political and economic problems have risen within the countries. Nobel Peace Prize writer Michael Spence predicts that as the trade with BRICS increases "the future of emerging economics is one of reduced dependence on industrial-country demand". Additional sources support this claim and suggest that the BRICS's problems may steam from a dependency on Global economies. This has caused a disheartening domestic demand and an increasing amount of displacements in the local market [9].

4 References

[1] Vijaykumar, N. (2011). Role of ICT in e-Governance: Impact of Cloud Computing in Driving New Initiatives.

[2] Grimsley, M., & Meehan, A. (2007). E-government information systems: Evaluation-led design for public value and client trust. European Journal of Information Systems, 16. 134-148.

[3] Kim, Gang-Hoon. Trimi, Silvana. & Chung, Ji-Hyong. Big Data Applications in the Government Sector.

[4] Quing, Zhao. Three Development Trends in e-Government: Cloud, Collaboration, and Big Data.

[5] Bhatnagar, Subhash. & Singh, Nupur. (2010). Assessing the Impact of E-Government: A Study of Projects in India.

[6] Miyata, Mayumi. (2011, September 23). Measuring impacts of e-government support in least developed countries: a case study of the vehicle registration service in Bhutan.

[7] Heeks, Richard. (2011, September 29). e-Government Benefits And Costs: Why e-Gov Raises Not Lowers Your Taxes.

[8] Burn-Murdoch, John. (2012, November 14). Government must improve big data strategy to realise savings of £33bn.

[9] Irwin, Neil. (2015, February 5). Global Debt Has Risen by $57 Trillion Since the Financial Crisis.

[10] Bello, Walden. (2014, September 4). The BRICS: Challengers to the Global Status Quo.

[11] Zizek, Slavoj. (2008). "Revolutionary Terror." In Defense of Lost Causes. London: Verso, 2008. 191. Print.

A Strategy for Multimedia Data File Retrieval using Hadoop Framework

Kyung Chang Kim and Jaekyung Lee
Department of Computer Engineering, Hongik University, Seoul, South Korea

Abstract - *The storage and retrieval of multimedia data is becoming increasingly important in many application areas including record management, video management and Internet of Things (IoT). In this paper, we propose a technique to retrieve a very large number of files, in multimedia format, using the Hadoop framework. Our strategy is based on the management of metadata that describes the characteristics of files that are stored in Hadoop Distributed File System (HDFS). To retrieve a file from HDFS, the metadata, in RDF format, of the file is looked up in MySQL database which then gives the file path to HDFS. Preliminary experiments on multimedia data files stored in HDFS shows the viability of the proposed strategy.*

Keywords: Multimedia data, record management, retrieval, Hadoop, metadata, MySQL

1 Introduction

In applications such as Internet of Things (IoT), video management system and record management system, large volumes of multimedia data are generated and should be managed. In order to record and manage these multimedia data, including text, document, image and video format, the size and scalability issues need to be solved. The multimedia data in above applications falls into the category of big data and the current research on big data mainly focus on analysis while less attention is paid to storage and retrieval.

Currently, the most commonly used framework to store and process big data is the Apache Hadoop[1]. Hadoop basically consists of Hadoop Distributed File System (HDFS)[2] and MapReduce[3]. In HDFS, the big data is stored in distributed manner which is then fed to MapReduce that processes the data in distributed manner. Hadoop also consists of a Namenode that contains metadata associated with the files in HDFS and Datanodes that contain actual files.

The metadata in the Namenode contain just basic information on the files and cannot be used for intelligent retrieval. For flexible retrieval of a very large number of files stored in HDFS based on various search criteria, a new metadata model is needed. In this paper, we propose a metadata model based on the RDF format where the metadata is stored in MySQL [4] database for flexible retrieval. We believe that the RDF format, which is the storage structure for the semantic web, is flexible enough to describe the characteristics of the files in HDFS.

The rest of the paper is organized as follows. We discuss related works in Section 2. In Section 3, we describe the metadata model based on the RDF format. Section 4 discusses the retrieval technique based on the proposed metadata stored in MySQL. Conclusion follows in Section 5.

2 Related Works

The Hadoop platform, which is the main platform for big data processing does not provide basic retrieval functions and uses libraries that support index and retrieval functions such as Lucene [5]. For example, Cloudera [6] distributes Cloudera Hadoop (CDH) that provides search functions in Hadoop using Lucene. However, it does not provide standardized index structures for search which means that the user has to develop own index structures when searching.

Recently, some research results have been reported to enhance the metadata in Hadoop to elevate file search. In [7], the enhanced metadata is represented using the relational model to overcome the limitations in Hadoop. In [8], a big metadata is proposed to manage the metadata in the big data environment.

3 Metadata Model

We first describe the characteristics of the various multimedia files. We then represent the detailed file characteristics in Resource Description Framework (RDF) [9], which is a Semantic Web data model. The metadata of the files, in RDF format, is then stored in MySQL database.

3.1 File Characteristics

Hadoop HDFS contains various multimedia files in video, audio, image and text format. Typical characteristics of a video file contain information like id, type, codec, width, height, playtime, frame, abitrate, akHz and achannel. For audio files, the information is id, type, playtime, bitrate, channel and kHz. For text files, the information is id, type and encoding. For image files, the information is id, type, width, height and bit.

3.2 RDF model for Metadata

The RDF model represent data as a labelled graph connecting resources and their property values with labelled edges representing properties. RDF structure is a set of triples in the form of < Subject, Property, Object >.

In order to use the RDF model to store the metadata of the files in HDFS, the RDF structure needs to be redefined. The Subject is the full path of the file, the Property is the various characteristics of the file and the Object is the values of the file characteristics. Table 1 shows an example of the metadata in RDF format

.

[Table 1] Example of Metadata using RDF

S	P	O
/test/video1	type	video
/test/video1	codec	avi
/test/video1	playtime	30
/test/video1	size	25
/test/video2	type	video
/test/video2	codec	wmv
/test/video2	width	640
/test/video2	height	360
/test/text1	type	text
/test/text1	codec	UTF8
/test/image1	type	image

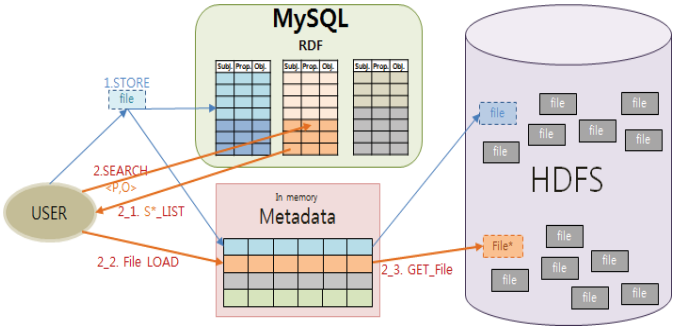
4 Retrieval Algorithm

The metadata represented in RDF format is first stored in MySQL database. When the user wants to store a file in HDFS, the metadata of the file is created and recorded in MySQL At the same time, the actual multimedia file is then stored in HDFS.

The metadata containing the file information can then be used to retrieve files using various search criteria. It is possible for the user to compose various queries targeted at the metadata stored in MySQL. Since Subject is the file path, Property is file characteristics, and Object is actual values of the file characteristics, the values of the Object can be used to query the MySQL database.

Figure 1 shows the file retrieval process. The user queries the MySQL database and finds the tuple corresponding to the metadata that satisfies the file search criteria. The query search condition is directed at the Property and Object columns. The Subject column of the retrieved tuple contains the path to the file in HDFS. The value of the Subject column is used to load the file from HDFS. Using this strategy, it is possible to retrieve multimedia files from Hadoop HDFS efficiently using various search criteria.

Preliminary experiments on various multimedia files stored and retrieved using the metadata stored in MySQL backs up the viability of our strategy.



[Figure 1] Retrieval Algorithm

5 Conclusions

Our paper proposes a new strategy based on metadata to efficiently retrieve a large number of multimedia files stored in Hadoop HDFS. We first defined the file characteristics of metadata for various multimedia files. The metadata of the files are represented in RDF format as the RDF model is very general and easy to express any type of data. The metadata, in RDF format, is then stored in MySQL database.

To retrieve a file from Hadoop HDFS, our strategy first gets the metadata tuple associated with the file by querying the MySQL database. The retrieved metadata tuple contains the path to the file in HDFS. Preliminary experiments show the viability of our strategy.

6 Acknowledgment

This work (Grants No. C0236971) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded by Korea Small and Medium Business Administration in 2015.

7 References

[1] Hadoop. http://hadoop.apache.org, 2014
[2] Shvachko, Konstantin, et al. "The hadoop distributed file system." Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on. IEEE, 2010.
[3] J. Dean, S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," In Proc. of the 6th Symposium on Operating Systems Design and Implementation, San Francisco CA, Dec. 2004.
[4] MySQL. http://www.mysql.com/, 2014.
[5] Lucene. http://lucene.apache.org/, 2014.
[6] Cloudera. http://www.cloudera.com/, 2014.
[7] Hakimzadeh, Kamal, Hooman Peiro Sajjad, and Jim Dowling. "Scaling HDFS with a Strongly Consistent Relational Model for Metadata." Distributed Applications and Interoperable Systems. Springer Berlin Heidelberg, 2014.
[8] Smith, Ken, et al. "Big Metadata: The Need for Principled Metadata Management in Big Data Ecosystems." Proceedings of Workshop on Data analytics in the Cloud. ACM, 2014.
[9] RDF Primer. W3C Recommendation. Feb, 2004 http://www.w3.org/TR/rdf-primer/

Big Data Analysis based Practical Applications in Construction

Yongho Ko[1], **and Seungwoo Han**[1*]
[1]Department of Architectural Engineering, Inha University, Incheon, South Korea

Abstract – *Big data analytic concept has changed fundamental paradigm of data analysis and prediction. Construction industry which seems to be the most conservative industry due to difficulty in data collecting, analyzing, and predicting is about to develop the advanced further phase with application of big data analytic methods. This study presents two cases which a big data analytic concept was adopted. The first case is for construction performance assessment and prediction. Home sales index prediction is the next. Database establishment with combination of simulation techniques and statistical approaches shows the fundamental ways how to use construction raw data stored in project management systems. Home sales index prediction based on search queries also suggests the effective methodology which enables users to investigate relevant environments and to produce reliable prediction results.*

Keywords: *construction performance, home sales index, prediction, assessment*

1 Introduction

The construction industry has been identified as one of the most conservative industries. It has been due to difficulties with emerging technologies applications in construction fields including house markets, where all conditions around each project have been totally different and various even very similar projects are operated. These difficulties have mostly been caused from limitations on the acquisition of data that could function as valuable input information on analytic and predictable methods [1]. For overcoming these limitations in construction, many studies related to searching efficient methods for collecting and analyzing informal data have been conducted in various fields in academia and industries. However, the effective ways for controlling and analyzing a huge amount of raw data which are collected in formal, informal, or semi-formal types have not suggested yet due to a sort of technical issue. The big data analytic concept significant findings in various fields currently get an attention as one of solutions for resolving these problems. The purpose of this study is to illustrate the preliminary cases illustrating how to apply the big data analytic concept to the construction. In accordance with this research objective, this study focuses two areas: 1) establishment of database for assessment and predictions of construction performances using data collected from construction sites [1, 2]; 2) home sales index prediction based on search query analysis. Figure 1 shows the diagram of two targeted areas in this study.

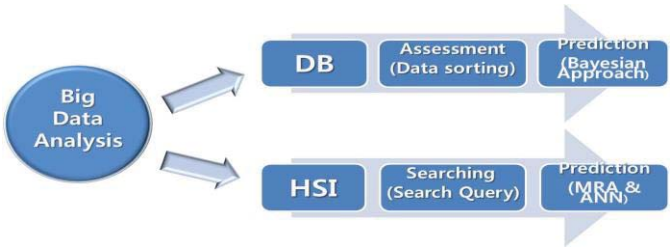


Figure 1. Two case studies for big data analytic concept applications

2 Construction Data

Difficulty in collecting precise raw data by consistent collection systems indicates why decision-making tools based on collected data have not been effective in the construction industry [1, 2]. This limitation in construction has brought other methodologies for data generations such as experimental designs based on statistic knowledge, and simulation techniques based on operational knowledge [1, 2]. Regarding predictions of construction performance, a multiple regression analysis presenting significant prediction results in various areas in academia and industry has been utilized as the representative methods based on the statistic knowledge. A fuzzy logic evaluating qualitative information to quantitative data and an artificial neural network also showing appropriate uses in prediction in industrial engineering have been mostly used based on artificial intelligences. However, it notes that these methods have not performed without collected raw data that reflects feasible cases under various conditions [1, 2].

3 Practical Applications

3.1 Database establishment for assessment and prediction of construction performance

Prediction of construction performance which is normally counted as fundamental criteria for making preliminary operational plans has been mostly studied since decision making tools were introduced in construction. Also precise predictions are critically demanding accurate assessment [1]. This case is for developing a database by using the simulation technique which has been recognized as an appropriate methodology for analyzing construction

performance but is limited in terms of its application to actual cases. The first phase of this database is to function proper assessment of construction performance data typically stored in project management information system as various types of data in formal, informal, and semi-formal formats. The second phase is then to predict performances using appropriate methodologies such as pattern recognitions and Bayesian approaches [2]. These two main phases in the database, assessment and prediction are conducted based on big data analytic methods. Figures 2 and 3 illustrate these two phases of output screen of the database [1, 2].

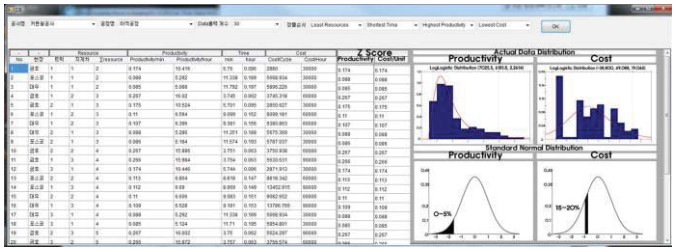


Figure 2. Output of the database for performance assessment [1]

NUM	SET	LEVEL	productivity
1	1	1	4.52
2	1	1	4.88
3	1	1	2.89
4	2	1	2.36
5	2	1	5.50
6	2	1	4.32
7	2	1	3.81
8	2	1	4.17
9	2	1	4.56
10	2	1	4.62
11	2	1	4.86
12	3	1	7.84
13	3	1	8.91
14	3	1	8.35
15	3	1	3.41

Figure 3. Output of the database for performance prediction [2]

3.2 Home sales index prediction

Home sales index has been identified to be one of key factors for establishment of long-term economic plans in government and of fundamental elements for preliminary design plans of construction projects. Subjective decision from empirical analysis and just experience of professionals were generally used in prediction of home sales index. This was derived from the difficulty and practical limitation posed by collection and analysis of a huge amount of relevant data. The case of home sales index prediction in this study utilizes the similar methods based on search queries in internet basis which was initiated by Google for detecting influenza epidemics [3]. This case shows a new home sales index prediction method based on a big data analytic concept implemented with statistical approaches such as cluster and principal component analysis. Multiple regression analysis and artificial neural network are applied for enhancing prediction results. Figure 4 illustrates the prediction results produced by these two methods.

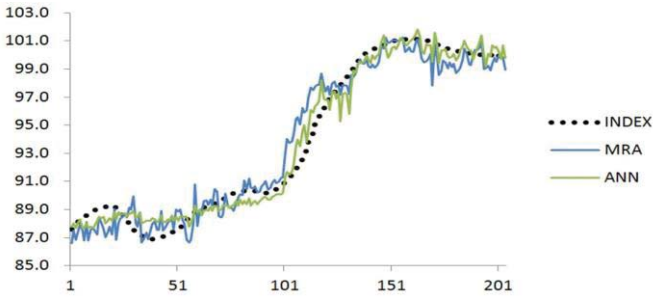


Figure 4. Prediction results

4 Conclusions

In construction, continuous efforts of development and practical application of appropriate decision making tools which were validated in other fields have been pursued. Several studies presented that big data analytic concept would be feasible for practical applications. This study suggested the practical cases for solving problems identified in the construction area especially practical applications. The cases are 1) construction performance assessment and prediction using raw data from project management information systems, and 2) home sales index prediction using search queries. This study presented the high possibility on a big data analytic concept of practical applications in construction with provision of significant findings and results. The methodologies and cases can be beneficial to other researchers interested in appropriate analytic and predictive methods in construction. It would be great initiations of the construction industry which make it more competitive against other manufacturing industries.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF), funded by the Korean government (NRF-2012R1A2A2A01046193)

5 References

[1] Seungwoo Han, Taehee Lee, and Yongho Ko. "Implementation of Construction Performance Database Prototype for Curtain Wall Operation in High-Rise Building Construction", Journal of Asian Architecture and Building Engineering, 13, 1, 149-156, Jan 2014

[2] Yongho Ko, and Seungwoo Han, "Development of Construction Performance Monitoring Methodology using the Bayesian Probabilistic Approach", Journal of Asian Architecture and Building Engineering, 143, 1, 73-80, Jan 2015

[3] Jeremy Ginsberg, Matthew H. Mohebbi, Rajan S. Patel, Lynnette Brammer, Mark S. Smolinski, and Larry Brilliant, "Detecting Influenza Epidemics using Search Engine Query Data, Nature, 457, 7232, 1012-1014, Feb 2009

SESSION

BIG DATA ANALYTICS AND RELATED ISSUES + LATE BREAKING PAPERS AND POSITION PAPERS

Chair(s)

TBA

Applying Fuzzy C-Means and Artificial Neural Networks into a High-Order Fuzzy Time Series Prediction Model

Mu-Yen Chen, and Hsiu-Sen Chiang
Department of Information Management, National Taichung University of Science and Technology, Taichung, Taiwan

Abstract - *A novel high-order fuzzy time series model for stock price forecasting is presented based on the fuzzy c-means (FCM) discretization method and artificial neural networks (ANN). In the proposed model, the FCM discretization method obtained reliable interval lengths. In addition, the fuzzy relation matrix was obtained from ANN, mooting the need for complex and time-consuming matrix operations. The proposed model was validated using experimental datasets from authentic university enrollment data. Empirical results indicate the proposed model outperforms existing methods for forecasting time series in terms of root mean squared errors (RMSE).*

Keywords: Fuzzy time-series, fuzzy c-means, artificial neural networks

1 Introduction

The 2007–2010 global financial crisis involved housing asset bubbles, credit booms, predatory lending, incorrect pricing of risk and the collapse of the shadow banking system. From January to August 2010, the US FDIC (Federal Deposit Insurance Corporation) (2015) announced 111 U.S. bank failures, and an additional 231 failures occurred over the next four years [1]. The crisis has had a significant negative impact on the United States and the European Union. From 2010 to 2012, four euro-zone countries (Greece, Ireland, Portugal and Cyprus) suffered continuous negative growth, leaving national governments unable to meet national debt payments [2]. Recently, Business Insider (2015) has also claimed that "the euro crisis is entering a new, highly dangerous phase, and once again Greece finds itself at the centre" [3].

Time series forecasting is used to predict future performance in many fields, facilitating the preparation for and mitigation of potential crises including economic shocks, power outages and droughts. Such forecasts are currently made using methods including regression analysis, moving average, autoregressive conditional heteroscedastic (ARCH) models [4], Generalized ARCH (GARCH) [5]. However, these statistical models are highly reliant on historical data which must follow a Gaussian distribution to optimize forecasting performance. Moreover, traditional time series forecasting models cannot deal with fuzziness or uncertain datasets because they lack an accurate measurement of certain data, or have difficulty obtaining actual measured values [6]. These issues are commonly addressed using fuzzy time series which can be applied to linguistic value datasets to produce accurate forecasting results. Fuzzy time series models have been used successfully many times to forecast nonlinear datasets in such widely varying applications as course enrollment [7], power loading [8], wind speed [9] and stock market performance [10, 11, 12, 13].

Following the most recent global financial crisis and European debt crisis, many researchers have urged the development of novel time series models for predicting economic events, such as stock market crashes, housing bubbles and credit booms. Currently, financial forecasting relies mainly on mathematical and statistical methods [14, 15, 16], and time series models [17]. Many theories and techniques for financial forecasting have been used in fundamental and technical analysis. This research applies fuzzy c-means (FCM) and artificial neural networks (ANN) to the construction of a high-order fuzzy time-series model for use with university enrollment datasets. Future research will aim to solve time series problems for financial predictions of stock market performance.

2 Literature Reviews

2.1 Fuzzy Time-Series Definitions

Song and Chissom [18, 19, 20] first proposed the application of fuzzy theory in time series in 1993 to deal with problems involving uncertain linguistic information. Fuzzy time series are designed using fuzzy sets [21, 22] and can thus overcome the disadvantage of traditional time series which can only deal with real numbers. Recently, Song and Chissom [18, 19, 20] have presented additional definitions for fuzzy time series as follows [20].

Definition 1. A fuzzy sets $A(t)$ in the universe of discourse U, where $U = \{u_1, u_2, \ldots, u_n\}$ can be represented as follows:

$$A(t) = f_{A(T)}(u_1)/\, u_1 + f_{A(T)}(u_2)/\, u_2 + \ldots + f_{A(T)}(u_n)/\, u_n$$

where f_A is the membership function of the fuzzy set $A(t)$ and $f_{A(T)} : U \rightarrow [0, 1]$, $f_{A(T)}(u_i)$ denotes the membership degree of u_1 in the fuzzy sets $A(t)$, $1 \leq i \leq n$.

Definition 2. Let $A(t)$, $A(t\text{-}1)$, $A(t\text{-}2)$,…,and $A(t\text{-}n)$ be the fuzzy sets in a time series. Assume that $A(t)$ is caused by $A(t$ -

1), $A(t\text{-}2)$,…,and $A(t\text{-}n)$. Then the nth-order fuzzy logic relationship can be represented as follows:

$$A(t\text{-}n), A(t\text{-}n+1), A(t\text{-}n+2), \ldots, A(t\text{-}1) \rightarrow A(t)$$

Definition 3. Let $A(t - 1) = A_i$ and $A(t) = A_j$, where A_i and A_j are fuzzy sets. The fuzzy logical relationship (FLR) can be denoted by $A_i \rightarrow A_j$, where A_i is called the left-hand side (LHS) and A_j is the right-hand side (RHS) of this relationship.

2.2 Fuzzy Time-Series Models

Over the past decade, several modifications to fuzzy time-series models have been proposed and can be categorized into high-order models [23, 24, 25] and artificial intelligence approaches. Because the first-order fuzzy time series model is established only using a simple fuzzy set structure, it has difficulty dealing with complex fuzzy logical relationships [26, 27]. For this reason, Own and Yu (2005) proposed a novel high-order fuzzy time series model to overcome complex time series issues [28]. Li and Cheng (2007) presented a deterministic high-order fuzzy time series model to forecast enrollment at the University of Alabama [29]. Their proposed forecasting model outperformed existing conventional models. Singh (2009) used many different orders as forecasting parameters and employed a w-step fuzzy predictor to test against datasets for University of Alabama student enrollment and Lahi crop production [30]. The following year, Li et al. (2010) developed a deterministic vector long-term forecasting (DVL) method, with performance evaluation by the Monte Carlo method with real datasets [25]. Chen et al. (2011) considered two factors to construct the high-order fuzzy logical relationship (FLR) groups [23]. To predict prices in the Taiwan stock market, Cheng et al. (2013) used the ordered weighted averaging (OWA) operator to integrate high-order data into the adaptive network-based fuzzy inference system (ANFIS) procedure [24].

For the artificial intelligence approach, Hsieh et al. (2011) integrated wavelet transformations and recurrent neural networks (RNN) to forecast international stock markets [31]. The same results were computed by RNN with Chandra and Zhang (2012) [32], and Smith and Jin (2014) [33]. Chen and Kao (2013) integrated particle swarm optimization and support vector machine techniques into their fuzzy time series model. Experimental results showed their proposed method outperformed existing conventional methods for stock market forecasts [34]. Recently, the same results were obtained that the support vector machine technique integrated with the proposed fuzzy time series models from Ruan et al. (2013) [35], Yang et al. (2015)[36] and particle swarm optimization approach with Pulido et al. (2014) [37], Singh and Borah (2014) [38], and Lin et al. (2015) [39].

2.3 Fuzzy C-Means Clustering Method

Bezdek's (1981) fuzzy C-means (FCM) algorithm [40] is the most popular classical fuzzy clustering method. FCM tries to divide the dataset by the minimizing the fuzzy least-square error objective function within the group with respect to the fuzzy membership u_{it} and center v_i:

$$J_\beta(X,U,V) = \sum_{i=1}^{c}\sum_{t=1}^{n} u_{it}^{\beta} d^2(x_t; v_i) \qquad (1)$$

wherein $\beta > 1$ for adjusting the noise in the dataset is the fuzziness index, n is the number of feature vectors x_t, $c > 2$ is the number of clusters in the dataset, and $d(x_t; v_i)$ is the measure of similarity between a datum and a center. In addition, the J_β minimizes the constraints as follows:

$$0 \le u_{it} \le 1 \qquad \forall i,t,$$
$$0 < \sum_{t=1}^{n} u_{it} \le n \qquad \forall i,$$
$$\sum_{i=1}^{c} u_{it} = 1 \qquad \forall t$$

generating a minimized pseudo-iterative algorithm known as the FCM algorithm. Each center v_i and the membership degree u_{it} are updated according to the following expressions.

$$v_{ij} = \frac{\sum_{t=1}^{n} u_{it}^{\beta} x_{kj}}{\sum_{i=1}^{n} u_{it}^{\beta}} \qquad (2)$$

$$u_{it} = \frac{1}{\sum_{j=1}^{c}\left(\frac{d(x_t; v_i)}{d(x_t; v_j)}\right)^{2/\beta-1}} \qquad (3)$$

where j is a variable on the feature space, i.e. j = 1,2,. . . ,m.

3 Research Methodology

The overall flowchart of the proposed model is shown in Fig. 1. The novel fuzzy time-series prediction model is built through two steps: pre-processing and optimization partitioning.

3.1 Pre-processing Phase

3.1.1. Data Source and Variable Selection

A single simple variable, "Actual", is used as the input variable to the proposed model to forecast student enrollment at the University of Alabama. Enrollment data from 1971 to 1992 are summarized in Table 1.

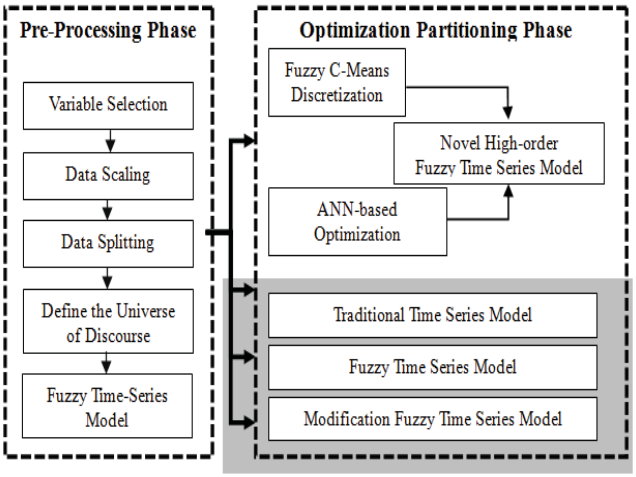


Fig.1. Research Methodology

Table 1. Enrollment data of University of Alabama

Year	Actual
1971	13055
1972	13563
1973	13867
1974	14696
1975	15460
1976	15311
1977	15603
1978	15861
1979	16807
1980	16919
1981	16388
1982	15433
1983	15497
1984	15145
1985	15163
1986	15984
1987	16859
1988	18150
1989	18970
1990	19328
1991	19337
1992	18876

3.1.2. Data Splitting

Most studies verify forecasting performance using the *data splitting* method wherein the experimental dataset is split into two subsets, one for training and one for testing. However, in this research, *data splitting* is not needed because the forecasting performance will be evaluated for each year in the enrollment data.

3.1.3. Define the universe of discourse, *U*

In the traditional fuzzy time series model, we first select two numbers D_{min} and D_{max} as the respective minimum and maximum values in the target dataset. We can thus determine the universe of discourse U as $[D_{min} - D_1, D_{max} + D_2]$, where D_1 and D_2 are two appropriate positive numbers. For example, based on enrollment data from 1971 to 1992, the U of the enrollment data from the University of Alabama are 13000 and 19400 according to a minimum enrollment of 13055 and a maximum enrollment of 19337, where $D_1 = 55$ and $D_2 = 63$. However, the FCM discretization method was applied to obtain reliable interval lengths, thus no measuring methods were needed to determine a suitable interval length for the discrete fuzzy sets [41].

3.1.4. Fuzzy Time-Series Model

Many previous studies followed Miller (1994) in establishing a universe of discourse U into seven intervals of equal lengths (i.e., u_1, u_2, u_3, u_4, u_5, u_6, u_7) because that human short-term memory function is seven, plus or minus two [42]. In this research, we also adopted the Miller's (1994) approach o establish the fuzzy sets A_1, A_2, …, A_7 of the universe of discourse U.

3.2 Optimization Partitioning Phase

3.2.1. FCM-based discretization partitioning

In this research, we adopted FCM to identify the optimization intervals based on Cheng's [43] model, which can adjust the linguistic interval and allow the observations to be more reliably fuzzified into the optimal linguistic values. This step includes two sub-steps:

Step 1. Applying time series *T* into *c* fuzzy clusters

Assume a time series T of u attribute with v observations. In this step, the appropriate fuzzy clustering method selects a cluster time series T into c ($2 \leq c \leq n$) cluster. By applying fuzzy clustering methods to the partitioning and fuzzifying process, each fuzzy cluster is used to predict the impact of fitness on the accuracy rate. In this step, FCM is selected because it is the most popular fuzzy clustering method. However, the optimal number of clusters remains an open and subjective question. Thus, following Miller (1994), seven clusters are used here to identify the cluster that corresponds to the limits of human perception.

As shown in Eq.(2), this step applies FCM to the time series T, and each clustering center is denoted as v_{ij} (i = 1,2,. . . , c and j = 1,2,. . . ,m). In advance, the Euclidean distance was used to measure the distance of each cluster iteratively. As shown in Eq.(3), the memberships u_{it} (t = 1,2,. . . ,n) of each

cluster center was computed when the cluster centers are shown to be stable.

Step 2. Fuzzifying the time series T(t) as fuzzy time series A(t)

As described in Step 1, the fuzzy time series {A(t) = [u_{1t},u_{2t}, . . .,u_{ct}]} could be caused by the time series *T*(t). In addition, this approach can easily determine whether A(t) belongs to cluster C_i when the maximum membership of A(t) occurs in cluster C_i.

Thus, using C_i as the center of C_i, each cluster is ordered according to its main attribute value rankings. The clusters are used to define the orderly linguistic variable L_r (r = 1,2,. . . , c). For instance, assume there are three clusters with centers of 40,120 and 80. Their centers are used to arrange them separately as C_1, C_3, and C_2, and define their corresponding linguistic variables as L_1, L_3, and L_2.

3.2.2. Artificial Neural Networks Optimization

This step includes two sub-steps used to verify our proposed model:

Step 1. Use artificial neural networks to construct the FLRs

When the time series model is complete, the fuzzy relationship between the different time orders can be deduced. The first-order FLR is always built from the previous fuzzy set $A(t\text{-}1) \rightarrow A(t)$. For example, in Table 2, $A(13867) \rightarrow A(14696)$ is a relationship, then a fuzzy relationship $L_2 \rightarrow L_3$ is used to substitute $A(13867)$ and $A(14696)$ with L_2 and L_3. Finally, the sequence of the FLRs is as follows: $L_3 \rightarrow L_4$, $L_4 \rightarrow L_4$, $L_4 \rightarrow L_4$, $L_4 \rightarrow L_5$, $L_5 \rightarrow L_6$, $L_6 \rightarrow L_6$ as shown in Table 2.

Table 2. Linguistic values of enrollment data

Year	Enrollment data	Linguistic values
1973	13867	L_2
1974	14696	L_3
1975	15460	L_4
1976	15311	L_4
1977	15603	L_4
1978	15861	L_5
1979	16807	L_6
1980	16919	L_6

We use a second-order fuzzy time series. For example, FLRs involves a structure composed of three consecutive fuzzy sets $A(t\text{-}2)$, $A(t\text{-}1) \rightarrow A(t)$. Following this concept, two input variables $A(t\text{-}2)$ and $A(t\text{-}1)$ were used as input values for artificial neural networks, and the use of these two input values, it can be assumed fuzzy time series $A(t)$.

Table 3. FLR construction for fuzzy time series model

Observation no.	*A(t-2)*	*A(t-1)*	*A(t)*	*Input-1*	*Input-2*	*Output*
1	-	-	L_1	-	-	-
2	-	L_1	L_2	1	-	2
3	L_1	L_2	L_3	1	2	3
4	L_2	L_3	L_4	2	3	4
5	L_3	L_4	L_5	3	4	5
6	L_4	L_5	L_6	4	5	6
7	L_5	L_6	L_7	5	6	7

This research set the linguistic value L_i of $A(t\text{-}2)$ and $A(t\text{-}1)$ as the input (*input-2* and *input-1*)variables for an artificial neural network. The linguistic value L_i of *the* $A(t)$ fuzzy time series can be seen as the forecasting value following computation using the artificial neural network. In Table 3, the input variables for the second observation [L_1] are 1, and then the output variable L_2 is 2. This FLR is known as the first-order fuzzy time series. Take the third observation as an example, the input variables [L_1, L_2] are 1 and 2, and then the output variable L_3 is 3. This FLR is known as the second-order fuzzy time series.

Step2. Forecasting results with defuzzification process

The forecasting values were obtained following defuzzification, and the results were computed using the artificial neural network outputs from Step 1.

4 Application to Enrollment Data

We collected twenty-two years of from the University of Alabama (1971-1992. We first define the universe of discourse *U* of the universe by a minimum enrollment number, and the maximum number for enrollment history for use in the initial iteration. FCM-based discretization partitioning was then applied to change the number of clusters between 5 and 20. We then calculated the appropriate FLR by using artificial neural network optimization recommendations. In constructing the fuzzy relationship, the number of neurons in the hidden layer is tuned between 1 and 10, with only one neuron in the input and output layers. Therefore, we implemented a hybrid prediction model with 160 architectures based on 10 different neural network architectures and 16 different clusters.

In this experimental design, the root mean square error (RMSE) value is used as the performance evaluation index by Eq.(4).

$$RMSE = \frac{\sqrt{\sum_{t=1}^{n} \left(actual(t) - forecast(t)\right)^2}}{n} \quad (4)$$

Enrollment forecasting was conducted using several popular models, including those proposed by Song and Chissom (1993b) [19], Song and Chissom (1994) [20], Sullivan and Woodall (1994) [44], Chen (1996) [45], and Cheng et al. (2008) [10]. As shown in Table 4, the proposed models significantly outperformed the traditional time series models.

Table 4. Results comparison

Methods	RMSE
Song and Chissom (1993b)	642.26
Song and Chissom (1994)	880.73
Sullivan and Woodall (1994)	621.33
Chen (1996)	638.36
Cheng et al. (2008)	478.45
Proposed method – first order	191.94
Proposed method – second order	174.03

5 Conclusions

A new high-order fuzzy time series model is proposed using the FCM algorithm to handle discrete partitioning issues, forecasting predicted values based on an artificial neural network optimization method. The proposed model outperforms five conventional fuzzy time series models for forecasting university enrollment.

This research presents a high-order fuzzy time series model to solve related time series forecasting problems. However, additional research is considered. First, the model should be applied to additional real-world datasets to provide more performance verification. Second, many curve fitting algorithms can be applied to predict the performance of neural network models. Finally, future researches could deal with discretization partitioning issues by using bio-inspired computing approaches, like particle swarm optimization (PSO), ant colony optimization (ACO), or artificial bee colony (ABC).

Acknowledgments

The authors thank the support of Ministry of Science and Technology (MOST) of the Republic of China (ROC) to this work under Grant No. MOST103-2410-H-025-022-MY2 and MOST103-2410-H-025-017. The authors also gratefully acknowledge the anonymous reviewers for their valuable comments and constructive suggestions.

6 References

[1] FDIC, Failed Bank List, 2015. Available from the website database: http://www.fdic.gov/bank/individual/failed/banklist.html

[2] CBS News, "Eurozone unemployment at record high in May", 2013. Available from the website database: http://www.cbsnews.com/news/eurozone-unemployment-at-record-high-in-may/

[3] Business Insider, The Euro Crisis Is Entering A New, Highly Dangerous Phase, 2015. Available from the website database: http://www.businessinsider.com/the-euro-crisis-is-entering-a-new-highly-dangerous-phase-2015-1

[4] R.F. Engle, "Autoregressive conditional heteroskedasticity with estimates of the variance of United Kingdom inflation", Econometrica, vol. 50, pp. 987-1007, 1982.

[5] T. Bollerslev, "Generalized autoregressive conditional heteroscedasticity", Journal of Econometrics, Vol. 31, pp.307–327, 1986.

[6] B. Möller, U. Reuter, Uncertainty forecasting in engineering. Berlin: Springer, 2007.

[7] S.M. Chen, N.Y. Chung, "Forecasting Enrollments of Students by Using Fuzzy Time Series and Genetic Algorithms", Information and Management Sciences, Vol.17, No.3, pp.1-17, 2006.

[8] R. Efendi, Z. Ismail, M.M. Deris, "A new linguistic out-sample approach of fuzzy time series for daily forecasting of Malaysian electricity load demand", Applied Soft Computing, Vol. 28, pp.422-430, 2015.

[9] J. Wang, S. Xiong, "A hybrid forecasting model based on outlier detection and fuzzy time series – A case study on Hainan wind farm of China", Energy, Vol.76, pp.526-541, 2014.

[10] C.H. Cheng, T.L. Chen, H.J. Teoh, C.H. Chiang, "Fuzzy time-series based on adaptive expectation model for TAIEX forecasting", Expert Systems with Applications, Vol. 34 , pp.1126-1132, 2008.

[11] M.Y. Chen, "A high-order fuzzy time series forecasting model for Internet stock trading", Future Generation Computer Systems - The International Journal of Grid Computing and eScience, Vol.37, pp.461-467, 2014.

[12] M.Y. Chen, B.T. Chen, "Online Fuzzy Time Series Analysis Based on Entropy Discretization and a Fast Fourier Transform", Applied Soft Computing, Vol.14, pp.156-166, 2014.

[13] M.Y. Chen, B.T. Chen, "A hybrid fuzzy time series model based on granular computing for stock price forecasting", Information Sciences. Vol.294, pp.227-241, 2015.

[14] E.I. Altman, "Financial ratios discriminant analysis and the prediction of corporate bankruptcy", Journal of Finance, Vol. 23, pp. 589–609, 1968.

[15] W.H. Beaver, "Financial ratios as predictors of failure", Journal of Accounting Research, Vol. 4, pp.71–111, 1966.

[16] R.C. West, "A factor-analytic approach to bank condition", Journal of Banking & Finance, Vol. 9, pp. 253–266, 1985.

[17] T. Bollerslev, "Generalized autoregressive conditional heteroscedasticity", Journal of Econometrics, Vol. 31, pp.307–327, 1986.

[18] Q. Song, B.S. Chissom, "Forecasting enrollments with fuzzy time series–Part 1", Fuzzy Sets and Systems, Vol. 54 , pp.1-9, 1993a.

[19] Q. Song, B.S. Chissom, "Fuzzy time series and its models", Fuzzy Sets and Systems, Vol. 54, pp.269–277, 1993b.

[20] Q. Song, B.S. Chissom, "Forecasting enrollments with fuzzy time series–Part 2", Fuzzy Sets and Systems, Vol. 62, pp.1–8, 1994.

[21] L.A. Zadeh, "Fuzzy Sets", Information Control, Vol. 8, pp.338-353, 1965.

[22] L.A. Zadeh, "The concept of a linguistic variable and its application to approximation reasoning - Part I", Information Sciences, Vol. 8, pp.199-249, 1975.

[23] S.M. Chen, C.D. Chen, "TAIEX forecasting based on fuzzy time series and fuzzy variation groups", IEEE Transactions on Fuzzy Systems, Vol.19, No.1, pp.1–12, 2011.

[24] C.H. Cheng, L.Y. Wei, J.W. Liu, T.L. Chen, "OWA-based ANFIS model for TAIEX forecasting", Economic Modelling, Vol. 30, pp.442-448, 2013.

[25] S.T. Li, S.C. Kuo, Y.C. Cheng, C.C. Chen, "Deterministic vector long-term forecasting for fuzzy time series", Fuzzy Sets and Systems, Vol. 161, pp.1852–1870, 2010.

[26] K.H. Huarng, T.H.K. Yu, "A Type 2 fuzzy time series model for stock index forecasting", Physica A, Vol. 353, pp.445–462, 2005.

[27] K.H. Huarng, T.H.K. Yu, "The application of neural networks to forecast fuzzy time series", Physica A, Vol. 363, pp. 481–491, 2006.

[28] C.M. Own, P.T. Yu, "Forecasting fuzzy time series on a heuristic high-order model". Cybernetics and Systems: An International Journal, Vol. 36, pp.705–717, 2005.

[29] S.T. Li, Y.C. Cheng, "Deterministic fuzzy time series model for forecasting enrollments", Computers and Mathematics with Applications, Vol. 53, pp.1904–1920, 2007.

[30] S.R. Singh, "A computational method of forecasting based on high-order fuzzy time series", Expert Systems with Applications, Vol. 36, pp.10551–10559, 2009.

[31] T.J. Hsieh, H.F. Hsiao, W.C. Yeh, "Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm", Applied Soft Computing, Vol. 11, pp.2510-2525, 2011.

[32] R. Chandra, M. Zhang, "Cooperative coevolution of Elman recurrent neural networks for chaotic time series prediction", Neurocomputing, Vol. 86, pp. 116-123, 2012.

[33] A. Mohammadzadeh, S. Ghaemi, "Synchronization of chaotic systems and identification of nonlinear systems by using recurrent hierarchical type-2 fuzzy neural networks", ISA Transactions, 2015. doi:10.1016/j.isatra.2015.03.016

[34] S.M. Chen, P.Y. Kao, "TAIEX forecasting based on fuzzy time series, particle swarm optimization techniques and support vector machines", Information Sciences, Vol. 247, pp. 62-71, 2013.

[35] J. Ruan, X. Wang, Y. Shi, "Developing fast predictors for large-scale time series using fuzzy granular support vector machines", Applied Soft Computing, Vol. 13, No. 9, pp. 3981-4000, 2013.

[36] C.Y. Yang, J.J. Wang, J.J. Chou, F.L. Lian, "Confirming robustness of fuzzy support vector machine via ξ–α bound", Neurocomputing, Vol. 162, pp. 256-266, 2015.

[37] M. Pulido, P. Melin, O. Castillo, "Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the Mexican Stock Exchange", Information Sciences, Vol. 280, pp. 188-204, 2014.

[38] P. Singh, B. Borah, "Forecasting stock index price based on M-factors fuzzy time series and particle swarm optimization", International Journal of Approximate Reasoning, Vol. 55, No. 3, pp. 812-833, 2014.

[39] L. Lin, F. Guo, X. Xie, B. Luo, "Novel adaptive hybrid rule network based on TS fuzzy rules using an improved quantum-behaved particle swarm optimization", Neurocomputing, Vol. 149, Part B, pp. 1003-1013, 2015.

[40] J. C. Bezdek, “Pattern recognition with fuzzy objective function algorithms”. 1981, NY: Plenum Press.

[41] E. Egrioglu, C.H. Aladag, and U. Yolcu, “Fuzzy time series forecasting with a novel hybrid approach combining fuzzy c-means and neural networks”. Expert Systems with Applications, Vol. 40, pp.854–857, 2013.

[42] G.A. Miller, “The magical number seven, plus or minus two: Some limits on our capacity of processing information”, The Psychological Review, Vol. 101, pp.343–352, 1994.

[43] C.H. Cheng, “Multi-attribute fuzzy time series method based on fuzzy clustering", Expert Systems with Applications, Vol.34, No.2, 1235-1242, 2008.

[44] J. Sullivan, W.H. Woodall, “A comparison of fuzzy forecasting and Markov modeling”, Fuzzy Sets and Systems, Vol. 64, pp. 279–293, 1994.

[45] S.M. Chen. “Forecasting enrollments based on fuzzy time series”, Fuzzy Sets and Systems, Vol. 81, pp.311–319, 1996.

A Study on Framework for Efficient Government R&D Program Feasibility Analysis Using the Bayesian Network and Big Data

JaeHyuk Cho[1]
[1]Division of Industrial R&D Feasibility Analysis, Korea Institute of S&T Evaluation and Planning, Seoul, S.Korea

Abstract - *Effective management of R & D performance measurement and evaluation in order to configure the details of the process by considering the sequence and interdependence, a performance evaluation of the factors that impact on other factors should be considered. In this dissertation, we present the Efficient R & D Performance Evaluation Framework Development and Decision Making System using Bayesian networks for the objectivity security and evidence base. Also, not only the related patents and theses data were collected but also the extensive trend data from public web site was collected and analyzed. Based on vast amounts of data which were collected rapidly, the accurate feasibility analysis used to find out the latest technology trends could also be useful for optimizing efficiency when analyzing. The national R & D program to improve efficiency and effectiveness through the enhancement of financial commitment, and whether the recommendations reflected the effectiveness of secure and efficient R & D performance evaluation framework for the development and implementation of decision support system will be studied*.

Keywords: Bayesian Network, Big Data, Feasibility Analysis, Government R&D Program, Performance Analysis

1 Introduction

A system that efficiently supports the performance management processes of program selection, budget allocation and performance evaluation is essential in supporting objective and systematic operations and management of research program. Especially, researches that objectively and quantitatively evaluate core competencies through performance evaluation systems and enhance program usefulness are necessary. Also, In the feasibility analysis of government R&D program, researchers generally reference the data submitted by the government or related public institutions or survey the related data to analyze the trend, level and the status of technology[1]. Patents and theses are generally utilized to analyze the technology trend for the feasibility analysis. However, before the actual publishing or listing, the time gap of 1~2 years is required so that such material are failed to reflect the newest data.

2 Previous Research

Using a Bayesian Network has the advantages of graphically specifying the relationships among nodes through a visual diagram. The accuracy in the prediction process of inference greatly increases with prior knowledge and conditional probabilities from numerous data. In this study, we use AHP and Bayesian Network as our method for analysis.

2.1 Bayesian Network

Using a Bayesian Network has the advantages of graphically specifying the relationships among nodes through a visual diagram. The accuracy in the prediction process of inference greatly increases with prior knowledge and conditional probabilities from numerous data. In this study, we use AHP and Bayesian Network as our method for analysis.

2.2 Performance Analysis

The existing performance indicators show a lack of analysis on the capability of a researcher who is participating in research program. For example, researchers' capabilities were measured with the number of SCI/E journal publications, for which the qualitative evaluation of the publication was evaluated through the number of times cited by others; however, still, there does not exist a measure to evaluate the overall research capability of a researcher. The Research program performances are homogenously evaluated through the number of publications or the impact factor, whereby limiting evaluations that are accompanied for field characteristics.

2.3 AHP[2] and Bayesian Network

This study use a weight calculated from AHP priorities. Probability values are assigned through the following assumptions. First, A weight is defined by how much the indicator influences the program. Second, Probability is defined for the number of events occurred. Thirdly, Impact values of risk exposure follow binary values of '0' or '1'[3].

Since the impact value of the risk exposure is represented by a binary value of '0' or '1,' risk exposure can be considered as a

probability value, thereby suggesting that weights can be used as probability values when it is regarded as risk exposure. We developed the performance analysis framework by connecting AHP with Bayesian Network to understand the relative importance and priority among main indicators. It is possible to catch relative importance and priority among them in dual comparison of AHP. Based on relative importance and priority among main indicators, we obtained weights for these. It is also from a dual comparison of AHP. Bayesian Network expresses the relationships among indicators using the result of AHP comparison.

2.4 Efficiency Analysis Using Big Data

Researches for the Big Data platform on the scientific and technological literatures have been performed to develop service models for Big Data analysis centric to the national R&D information, to support the establishment of science and technology strategy and to minimize possible risks from the large scaled investment and the low success probability on such activities. However, no measure to scale the technology impact of web data like the quotation degree of patents and theses has been provided.

3 Feasibility and Performance Analysis

I can elicit performance analysis index using the weights we obtain as shown above, which are effective for comparing the quantitative performance of each program. Existing analysis is a plain method irrespective of priority between indicators. To overcome this limitation, the method using performance index can get high efficiency

3.1 Search Keyword Selection

I examine the level of impact of the core technologies of the feasibility study cases versus other technologies in relevant fields.

3.2 Data Collection using Web Crawler

It is expected that web resources generally are not appropriate to utilize for a feasibility study because the credibility of web resources in general has been lower than objective measurements such as theses and patents.

3.3 Results of Analysis

This dissertation used a real dataset such as theses, patents and trend data in order to analyze the level of technology impact of the core technology of the program subjected to a feasibility analysis.

3.4 Inference in Bayesian Network

Inference in Bayesian Network is to draw a value one wants by using variables that one already has. One of the most common methods is called variable elimination. It eliminates irrelevant variables by distributing the sum over the product.

3.5 Likelihood Function

All tables must be numbered consecutively. Table headings should be placed above the table. Tables should be as close as possible to where they are mentioned in the main text. Tables can span the two columns if need be within the page margins.

3.6 Posteriori Inference

I can infer posteriori probability using prior probability and likelihood function. When applying data, the prior probability uses weight and likelihood function observes data. Likelihood infers posteriori probability by estimating the probability between data that belong to the same population. The posteriori probability can figure out the weakness of performance indicator, as it can predict the result that appears T or F in advance.

4 Conclusions

In this dissertation, we selected main performance indicators and weighted them differently considering the relative importance and priority. This was to overcome the limitations of indicators and to analyze performances quantitatively as well as qualitatively. In addition, we suggested a performance analysis framework to infer posteriori probability. It resulted from our attempt to combine AHP with Bayesian Network. Through this result, we can provide valuable feedback to the researchers of program that are in the dissatisfaction realm. Also, the web crawler based on Google Web Search API is developed to enable the collection and analysis of the newest credible data from the internet space through Big data and technology impact index to reflect the technology impact on the newest trend data is proposed.

5 Acknowledgements

This research was supported by the research program of KISTEP(Korea Institute of S&T Evaluation and Planning)

6 References

[1] Korea Institute of S&T Evaluation and Planning, A Study on Standard Guidelines for Preliminary Feasibility Study on R&D Program(1th Edition), 2011

[2] Saaty, The Analytic Hierarchy Process, McGraw Hill, 1980

[3] Jae-Hyuk Cho, A Study on Framework for Effective R&D Performance Analysis of Korea Using the Bayesian Network and Pairwise Comparison of AHP, Journal of supercomputer, 2013

SESSION

ALGORITHMS AND APPLICATIONS FOR BIG DATA

Chair(s)

TBA

Performance of an Approximate, Multidimensional Range Index for Big Data

R. J. Wroblewski
SSC Pacific, San Diego, CA, USA

Abstract *- A common data-processing problem is the retrieval or counting of objects in a data store whose extent includes a given point or extent such as finding overlapping areas or volumes. Hierarchical-tree methods in common use are not well suited for dealing with this problem in distributed data stores. This paper explores some performance characteristics of an approximate, multidimensional range index. The approximate searches significantly reduce the number of out-of-range results generated by the multidimensional-indexing method, at the cost of missing some correct overlaps. Using simulations, we find good performance for modest search approximations. The best performance is achieved when the average extents are small in comparison to the scale of the dimension. The most interesting result is that the performance improves with increasing dimension.*

Keywords: range indexing, overlap indexing, multidimensional indexing

1 Introduction

A common data-processing problem is the retrieval or counting of objects in a data store whose extent includes a given point or extent. The extents can be one-dimensional, such as temporal durations, or higher-dimensional, such as areas, volumes, etc. The problem is an old one and has been approached in many ways: bounding lists, multidimensional indexing, quad-trees and the ilk, various hierarchical trees such as interval trees and R-trees, and simplex range searching [1-6]. The problem is considered solved for non-distributed data stores [7].

However, in the age of big data and distributed stores, this is no longer the case. Hierarchical trees are not well suited for key-value databases nor Hadoop disk dumps. Adapting R-trees for this environment is an area of ongoing research [8, 9].

The older method of multidimensional indexing has the advantage of suitability as part of the key in a NoSQL DB and is amenable to filtering via map/reduce processing. In this paper, we will look at the performance characteristics of an approximation to range searching with a multidimensional index.

2 Multidimensional Range Index

To start with, consider the one-dimensional range problem. Assume we have a collection of many ranges, characterized by a start and end point. We are given a test point and need to find what ranges from our collection overlap this point. There are three cases possible for each range: the point is before, during, or after, as illustrated in Figure 1.

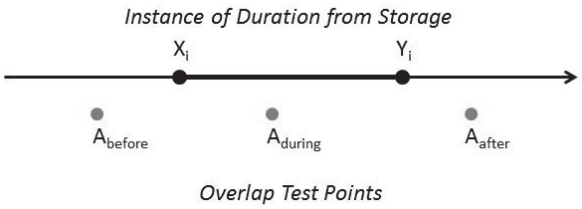


Figure 1: Overlap of One-Dimensional Ranges with Points.

Treating the start (X) and end (Y) points of the ranges as independent variables, we can plot the collection of ranges as a collection of points in a two-dimensional range space. The space divides into two regions with all the physically possible durations existing in one half, with points (zero-duration) existing on the dividing line. This is illustrated in Figure 2.

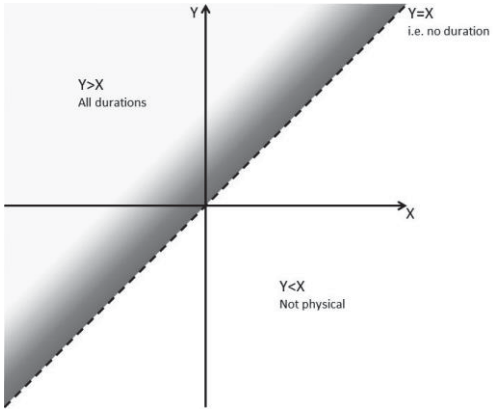


Figure 2: One-Dimensional Ranges Plotted in Two-Dimensional Space.

The test point A splits the range space into three distinct regions, as shown in Figure 3, that can be identified as the before, during, and after cases. Hence, an orthogonal search box with limits of [min(X), A] in the lower left to [A, max(Y)] on the upper right will find all durations that include A. Finding the befores or afters are simple one-dimensional searches using the upper- and lower-bounding lists.

Extending this to orthogonal searches in higher dimensions is straight forward: areas embed into a four-dimensional space, volumes embed into a six-dimensional space, etc. The search boxes become hypercubes. Combinations of before/during/after regions are all also higher-dimensional and hence are no longer amenable to the simple one-dimensional upper- and lower-bounding lists.

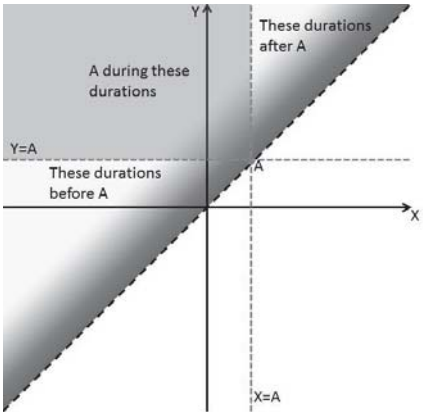


Figure 3: Location of Overlap Regions for Points.

The situation gets more complicated when the test object is also a range. There are now a number of different types of overlaps plus the before and after cases. Figure 4 summarizes these for the one-dimensional case. Note that the instances pulled from the data collection are displayed towards the bottom of this Figure.

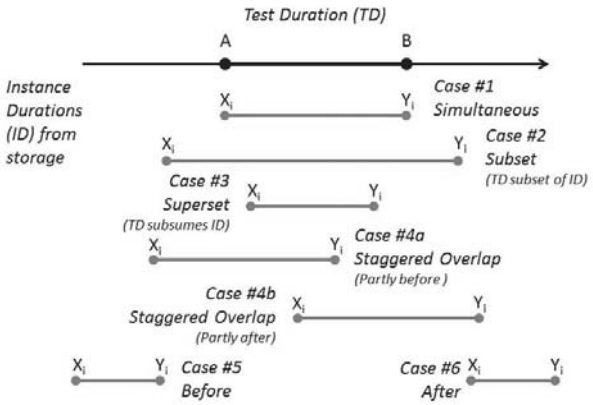


Figure 4: Overlap of One-Dimensional Ranges.

Concomitantly, the range space is broken up into more distinct regions, as drawn in Figure 5. As before, the befores and afters can be found directly using upper- and lower-bounding lists for this one-dimensional case. The rest of the regions require search boxes. The interesting search case is to determine if there are any overlaps. For a test interval [A, B], the search box has a lower-left corner of [min(X), A] and an upper-right corner of [B, max(Y)]. The reflection point of the test interval, [B, A], is the lower-right corner. Also as before, this can be generalized to extents of higher dimension with complexity increasing combinatorically.

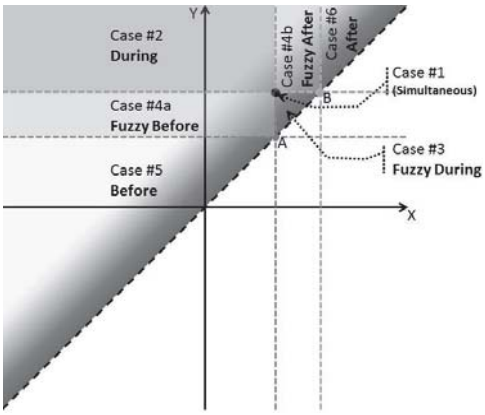


Figure 5: Location of Range Overlaps in Two-Dimensional Space.

One method of searching these range-space boxes is via multidimensional indexing (MDI) based on space-filling curves. These techniques have been around for fifty years [10, 11]. Additionally, they currently being successfully used as the key in NoSQL databases [12].

Space-filling curves have the property of linearly ordering multidimensional data such that points near one another in the multidimensional space are near in the linear ordering, *on average*. The key phrase is 'on average', which for Morton, Hilbert, or Gray-Code types means within cascading powers of two in each dimension. We will look at the Morton, or Z-order, index, which involves interleaving the bits of the binary representations of the dimensions and is computationally trivial to extend to higher dimensions.

To search using MDI you compute the indexes for the two points with all the lowest ranges and the highest ranges. This gives you the search limits in the linear index space. The issue with this search is that the index run between the limits can make large excursions outside of the desired search box. This is particularly exacerbated with higher dimensionality and is caused by the search box crossing bit boundaries, sometimes called fault lines. These excursions can be reduced significantly by splitting the search box.

The search box is split along the largest bit boundary, which is found by determining the largest bit of the XOR of the start and end indexes of the box. Figure 6 illustrates this process. In a) is shown a sample search box overlaid upon a Morton-indexing grid. The next sub-figure, b), highlights the run of the search indexes that cover this box. Computing the largest bit boundary crossing the box, c), gives us the split. Sub-figure d) illustrates the reduced index runs of the split search box, highlighted in a darker gray. The process continues in sub-figure e), which shows the independent splitting of the two search boxes along their major bit boundaries. Finally, the further-reduced index runs are highlighted in black in sub-figure f). This process of splitting the search boxes can be continued until there are no points exterior to the desired search box, i.e. no false alarms. Note that all points below and to the left of the lower bound and all points above and to the right of the upper bound of the search box are excluded from the excursion space.

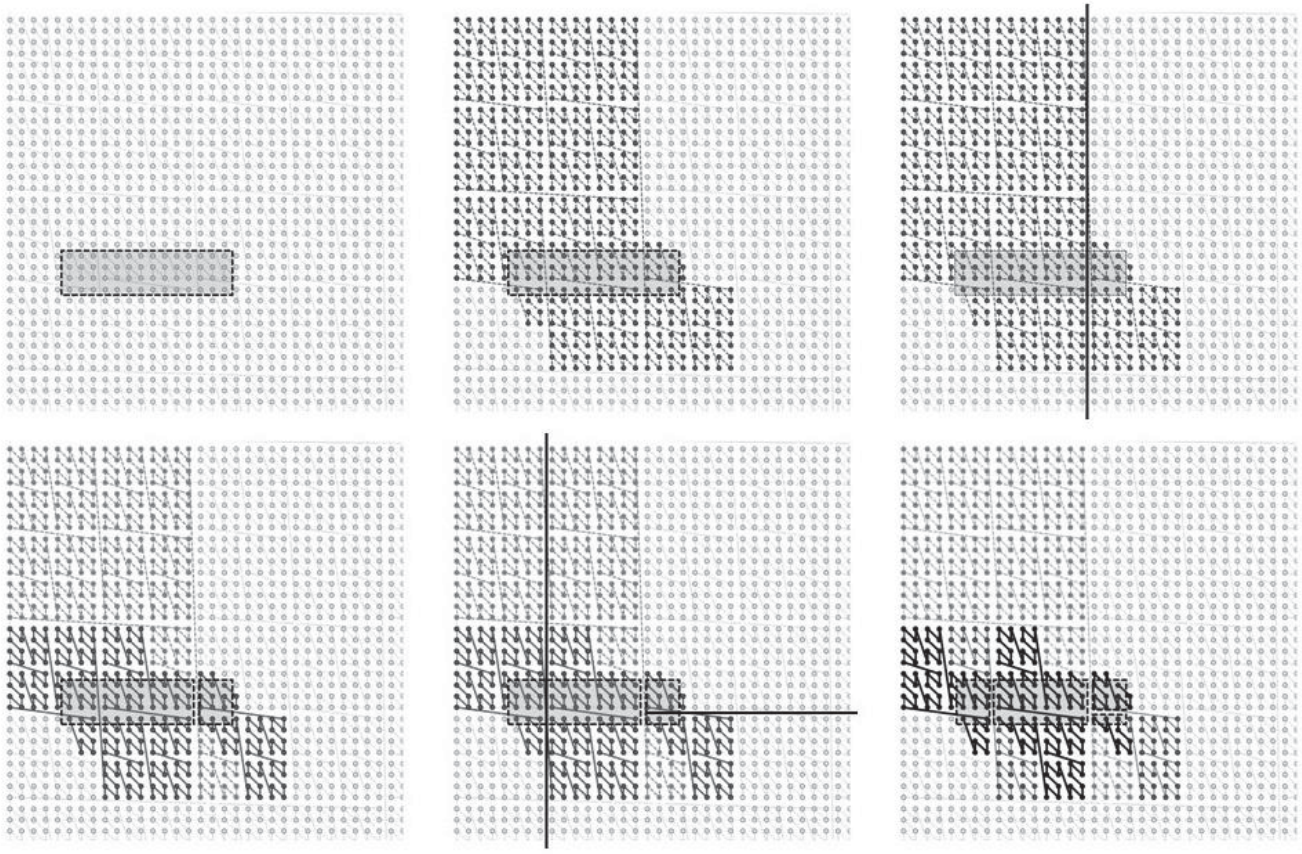


Figure 6. Illustration of Successive Search-Box Splitting Along Bit Boundaries.

It is straightforward to apply MDI to range spaces in one or more dimensions. The search boxes are well-defined, but have range limits that are either the minimum or maximum values in the data set. Although it is easy to keep track of these values on the fly, this means that on average, the search boxes will encompass half of the data set. The index runs will be significant, even with multiple splittings. Extracting a large fraction of a data store does not scale well to big data. This is not an unexpected result as even the exact answer will span a large fraction of the data set in the case of evenly distributed starts and ends.

However, in many realistic cases, the data will not be evenly distributed in the allowed range space. Average extents that are small compared to the scale of the dimension are typical. Consider, for example, tract-housing plot sizes compared to city or state scales, or the duration of an appointment compared to the length of the appointment book. It is not unreasonable then to model the ranges as an arbitrary start point within the scale, but with extents that are Gaussian distributed with a σ less than, and perhaps significantly less than, the size scale of the data. The perceptive reader will have noticed that the range plots have been suggestively shaded to indicate such a distribution.

This situation can be exploited using approximate search boxes that are significantly smaller than the exact solution yielding MDI searches with much better performance. The tradeoff is that this is not an exact

solution; some overlapping ranges will be missed. Decreasing the search-box size reduces the number of false positives coming from the index-run excursions, but at the cost of a higher number of missed solutions. This is a theme familiar to those versed in detection theory, and we will couch our performance results in terms of probability of detection, P_D, versus the probability of a false alarm, P_{FA}.

Figure 7 illustrates the search-box approximation and the capture of most of the desired overlaps. The search box has a lower limit at [A-$f_s\sigma$, A] and upper limit [B, B+ $f_s\sigma$], where f_s is a search-box size parameter. Typically, the missed detections will be those overlapping the test point or range near the edge of the extents. Less likely are the cases where the stored extents are fairly large. In the latter case, if important to a problem, these would generally be few enough to handle in a separate table. The false alarms (FA) can only exist in two, much-smaller triangular regions.

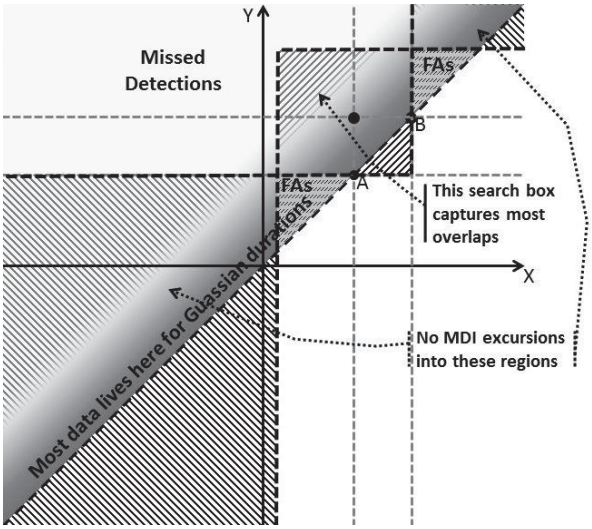


Figure 7: Location of Ranges with Gaussian-Duration Approximation.

3 Big-Data Considerations

In the world of distributed big data, Hadoop and Map/Reduce are the paradigms-du-jour. In the Hadoop Distributed File System (HDFS), data is distributed over many nodes and replicated for robustness [13]. The driving factor in the efficiency of the Hadoop system is the cost of seek times versus read times for physical-disk drives. Dumping a large block of the disk store into memory and filtering there with map/reduce functions is faster than directly accessing the data with numerous disk seeks [14].

Databases built on top of HDFS are generally non-relational, using some form of a key-value store. These NoSQL stores eschew typical database normalizations to take advantage of Hadoop's efficiencies by distribution; replicating data stored directly with the key, as opposed to pointing to another table and hence having another disk seek. One such database is Apache Accumulo, originally built by NSA along the lines of Google's Big Table, but with row-level access controls [15]. Accumulo distributes the data roughly along the lines of an ASCII sort of the key. This allows for Hadoop-efficient retrieves with map/reduce filtering.

For purposes of this discussion, we will take Accumulo as our database model. We imagine data being stored into the database using a multidimensional range index as the key. (That is, the row-id part of the key.) When a search is required, we use the approximate search box, described previously, and perform search-box splitting to reduce the false alarms.

However, there is a cost associated with reducing the false-alarm rate by splitting. Each search run corresponds to requesting a seek in the Hadoop store, and each seek results in a data dump of some size, S_{Hadoop}, typically 64 MB. This data is processed by map/reduce to extract the desired data. At some level of splitting, this becomes more expensive than just filtering the data with false alarms using map/reduce. A simplistic way to determine the optimal splitting is to look for the break-even point of data dumped. Ignoring issues such as the data density as a function of index and how many Hadoop seeks are required for long index runs, the splitting criteria is given by Equation 1.

$$P_{FA} N_{Records} S_{Record} \geq n_{splits} S_{Hadoop} \qquad (1)$$

Here the number of records due to false alarms is the probability of a false alarm, P_{FA}, times the number of records in the database, $N_{Records}$. The size of the false-alarm data depends on the average size of a record, S_{Record}. So long as this is greater than the data dump due to n_{splits} search boxes, we may split some more. Rearranging, we find the optimization criteria given in Equation 2:

$$\frac{n_{splits}}{P_{FA}} \leq \frac{N_{Records} S_{Record}}{S_{Hadoop}} = \frac{S_{Database}}{S_{Hadoop}} \qquad (2)$$

Noting that P_{FA} is a function of the number of splits, we have a fairly non-linear optimization equation. We find that the limiting parameter is the number of Hadoop-seek dumps it takes to dump the full database. For a 1 TB database this number is 16K, assuming a typical S_{Hadoop}. So for high false-alarm rates, we need to split the search boxes many times to minimize dumping a large fraction of the database. While on the other hand, low false-alarm

rates would need just a few levels of splits to reach a Hadoop-optimal search.

4 Simulations

To quantify this result, a number of simulations were run and analyzed. One-, two-, and three-dimensional overlaps were considered, corresponding to two-, four-, and six-dimensional range indexes. For each overlap case, three Gaussian-duration parameters were used: σ/Scale = 0.01, 0.03, and 0.1. (Each start and end point is chosen to be an integer between 0 and Scale = 100,000.) For the higher-dimensional overlap cases, the same σ was used for all dimensions. Figure 8 shows an instance of a one-dimensional case with σ/Scale = 0.01 and 10,000 ranges.

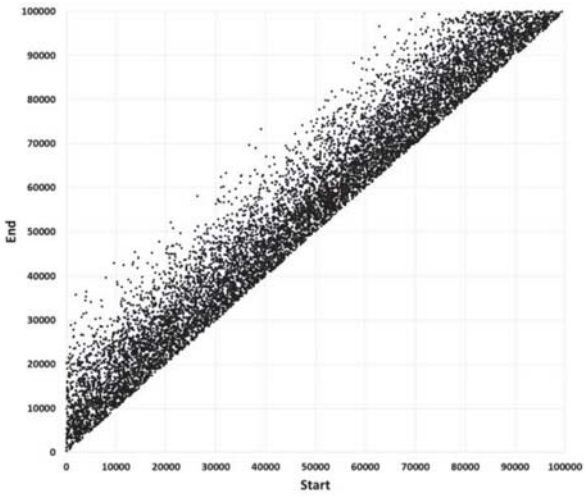


Figure 8: An Instance of a One-Dimensional Ranges, Gaussian-Distributed (σ = 0.1*Size Scale)

The actual instances of the overlap-range sets used for our results consisted of 100,000 points and eight separate instances were generated for each case. Hence, the sample space supporting the results is 800,000. Against each of these instances, a set of eight random test ranges was also generated. For each of these test ranges an exact solution of overlapped ranges from the simulated range-set instance by a brute-force method. Then the solution from the approximate multidimensional-overlap-index method was computed using search-box sizes of 1.5σ, 2.0σ, 2.5σ, 3.0σ, and 3.5σ. From each of these, the number of missed overlaps and the number of false alarms were tallied. The simulation and indexing codes were written in Ruby.

5 Performance Results

One way of analyzing and displaying performance of detection results is with a receiver-operating-characteristics (ROC) curve [16]. The probability of detection, P_D, is plotted versus the probability of false alarm, P_{FA}. Performance is better the further the curve is to the upper left, that is, higher P_D and lower P_{FA}.

Figure 9 shows the results. As expected, increasing the search-box size reduced the number of missed detections and increased the number of false alarms. For all of these cases, the P_D was over 0.9, and for many it was over 0.99. The latter corresponds to search-box sizes of 2.0σ and larger. The P_{FA} varied from 0.0006 to 0.13. The results shown correspond to choosing the optimal splitting level as discussed previously.

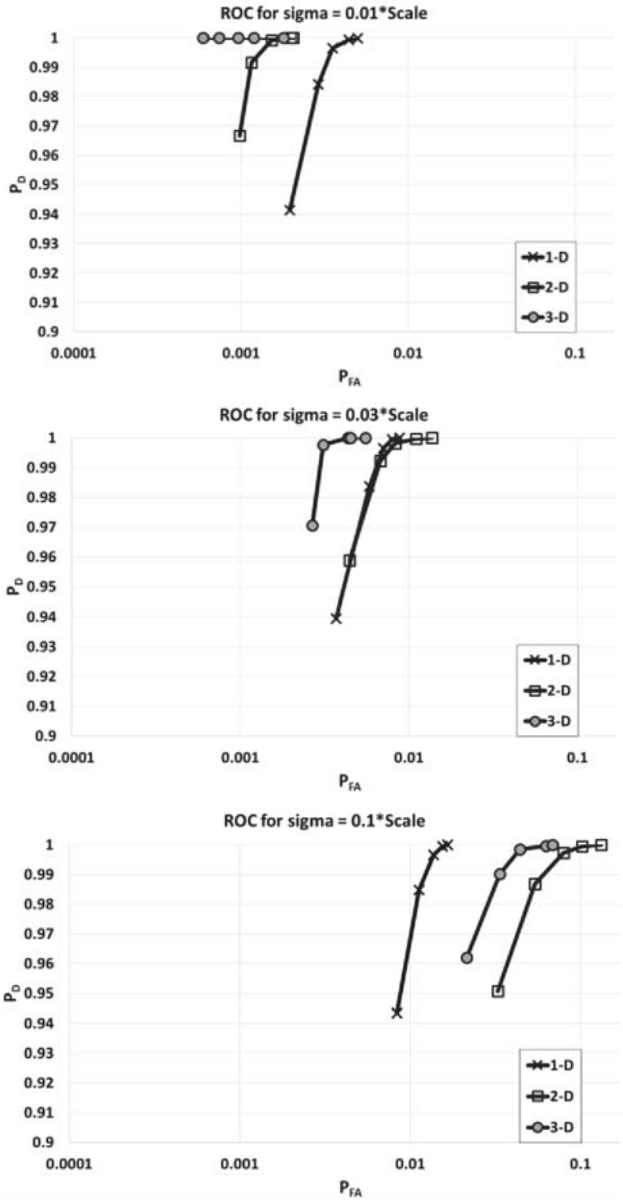


Figure 9: Performance of Multidimensional Overlap Indexing for One-, Two-, and Three-Dimensional Overlaps.

Also as expected, as the spread of the extents increases, the performance of the searches degrades. This is seen with the curves sliding to higher P_{FA}s going from Figure 9a to 9c. The dramatically high P_{FA}s occur for the largest search boxes (3.5σ) when σ=0.1*Scale. That is, the search box is over a third of the Scale.

What is interesting and unexpected is that for the smallest relative σ (=0.01*Scale), the performance *improves* with increasing dimension. The trend is seen starting to reverse at σ=0.03*Scale, but the three-dimensional results are still the best.

An important parameter in the performance is the size of the excursions of the index run outside of the search box. This is directly related to the P_{FA} and should be increasing with increasing dimension. To examine this we have calculated an excursion parameter, $S_{Excursions}$ = (length of index run / size of search box) – 1, which gives us the relative size of the excursions. Figure 10 shows this plotted against the P_{FA} for the cases shown above. Here we see that the excursions do indeed get significantly larger with increasing dimension in all cases. The increase of the excursions is somewhat exaggerated by the fact that fewer splits were performed on the higher dimensions because of the lower P_{FA}s.

So the performance improves despite the increasing excursion sizes for the smallest relative σ. This means that the excursions are encountering a lower population in the excursion region. Ah, the 'curse' of higher dimensionality. Although the excursion region is increasing on the order of 2^m with dimension, m, the probability of finding an object there is decreasing much faster for the tight distribution.

6 Summary and Conclusions

In this paper we have explored the use of multidimensional indexing in range space as a key with distributed data stores for efficient discovery of overlaps. The key innovation is the reduction of search-box sizes based on a not-unreasonable assumption of the data distribution. This assumption is that the extents of the data objects is small in relation to the scale of the dimension in the data.

In order to quantize the performance, a series of simulations was run, involving millions of random extents in one, two, or three dimensions, over three distribution scales, and five search-box sizes. Missed detections and false positives were tallied for all the cases. Searches were optimized by choosing the best splitting based upon HDFS parameters.

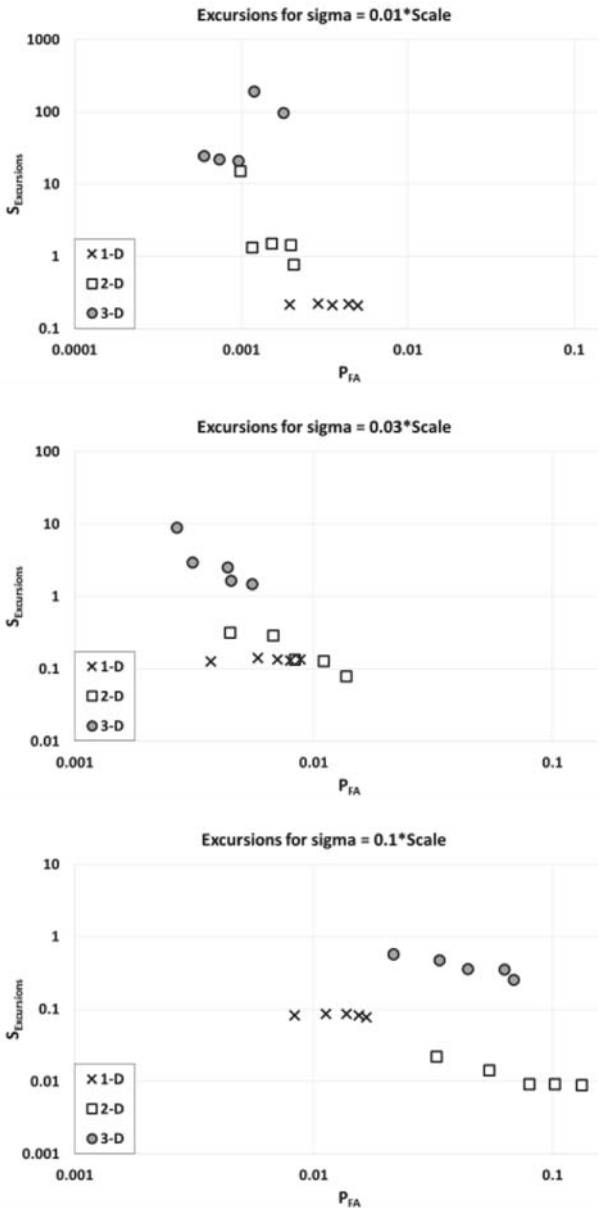


Figure 10: Relative Excursion Size vs. P_{FA} for One-, Two-, and Three-Dimensional Overlaps.

We found that the performance improved as expected for tighter distributions. Also as expected, decreasing the search-box size reduced the number of false positives at the cost of increasing the number of missed detections. A reasonable search-box size is around 2σ to 2.5σ with missed-detection rates lower than 0.01.

The most interesting, and unexpected, result is that for the tightest distribution, performance improved with increasing dimension. This is true despite the fact that the excursions outside the search box using the MDI method increases dramatically. We ascribe this result to the 'curse' of higher dimensionality, where, for these tight distributions, the probability of the excursion space being

populated decreases substantially faster than the increase of that space as the dimensionality increases.

7 Acknowledgements

The author would like to thank the Office of Naval Research (ONR) for financial support. We thank Scott McGirr for paper editing. This paper is the work of US Government employees performed in the course of employment and copyright subsists therein.

8 References

[1] Floreza, O.U., X. Qia, & A. Ocsab, "MOBHRG: Fast K-Nearest-Neighbor Search by Overlap Reduction of Hyperspherical Regions," ICASSP 2009, IEEE

[2] Hoelt, E.G., "A Qualitative Comparison Study of Data Structures for Large Line Segment Databases," 1992 ACM SIGMOD

[3] Orenstein, J., "A Comparison of Spatial Query Processing Techniques for Native and Parameter Spaces," 1990 ACM

[4] Keller, O., T. Kopelowitz, M. Lewenstein, "Range Non-overlapping Indexing and Successive List Indexing," Algorithms and Data Structures, 10th International Workshop, WADS 2007: 625-636, 2007

[5] T. Sellis, N. Roussopoulos, & C. Faloutsos, "The R+-Tree: a dynamic index for multi-dimensional objects," In Proc. 13th International Conference on VLDB, pages 507–518, England, September 1987

[6] de Berg, M., M. van Kreveld, M. Overmars, & O. Schwarzkopf, *Computational Geometry*, Second Revised Edition. Springer-Verlag 2000, Sec. 10.1: Interval Trees, pp. 212–217.

[7] Agarwal, P.K. & J. Erickson, Geometric Range Searching and Its Relatives, Contemporary Mathematics, 2004

[8] Liao, H., J. Han, & J. Fang, "Multi-dimensional Index on Hadoop Distributed File System," 2010 Fifth IEEE International Conference on Networking, Architecture, and Storage

[9] Francis, F. S. & P. Xavier, "Amendments to the R*-Tree Construction Principles in Distributed Environment," Int'l. Jour. of Engin. & Tech., ISSN: 2278-0181, Vol. 3, Iss. 5, 2014 May.

[10] Morton, G, "A Computer-Oriented Geodetic Data Base and a New Technique in File Sequencing," Tech Report, IBM Ltd., Ottawa, Canada, 1966.

[11] Asano, T., T. Roos, P. Widmayer, & E. Welzl, "Space-Filling Curves and Their Use in the Design of Geometric Data Structures," Proc. 2nd Latin Amer.Sympos. Theoret. Infomatics, Lecture Notes Comput. Sci., Vol 911, Springer-Verlag, 1995, pp. 36-48.

[12] Fox, A., C. Eichelberger, J. Hughes, & S. Lyon, "Spatio-temporal Indexing in Non-relational Distributed Databases", ISBN 9781479912933, 2013 IEEE

[13] EMC Education Services, *Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data*. John Wiley & Sons. 2014-12-19. p. 300. ISBN 9781118876220.

[14] Apache Hadoop MapReduce http://hadoop.apache.org/mapreduce/

[15] Apache Accumulo: http://accumulo.apache.org

[16] Van Trees, H. L., *Detection, Estimation, and Modulation Theory, Part I*, Wiley-Interscience, John Wiley & Sons, Inc., New York, 2001, pp. 36-46.

From Events to Local Dependence between Random Units in Big Date

B. Dimitrov
Department of Mathematics, Kettering University, Flint, Michigan, USA

Abstract - *In this article we show how some known to us measures of dependence between random events can be easily transferred into measures of local dependence between random variables. This enables everyone to see and visually evaluate the local dependence between uncertain units on every region of their particular values. We believe that the true value of the use of such dependences is in applications on non-numeric variables, as well as in finances and risk studies. We also trust that our approach may give a serious push into the microscopic analysis of the pictures of dependences offered in big data. Numeric and graphical examples should confirm the beauty, simplicity and the utility of this approach.*

Keywords: *local measures of dependence, regression coefficienstins, correlation, mapping the local dependence, big data tools, microscopic analysis of dependence*

1 Introduction

The Big Data is full of multi-dimensional simultaneous observations. This fact offers plenty of opportunities for establishing possible dependences between observed variables. Most of these dependences will be of global nature. However, there exist (or can be created) techniques to take a look on more details into it. In this talk we want to show the ideas of these microscopic looks.

The concepts of measuring dependence should start from the very roots of Probability Theory. Independence for random events is introduced simultaneously with conditional probability. Where independence does not hold, events are dependent. Further, the focus in books is on the independence. No textbook discusses what to do if events are dependent. However, there are ways to go deeply in the analysis of dependence, to see some detailed pictures, and use it later in the studies of random variables. This question is discussed in our previous articles [2] and [3]. Some particular situations are analyzed in [4] and [5]. We refer to these articles for making a quick passage to the essentials.

First we notice here that the most informative measures of dependence between random events are the two *regression coefficients*. Their definition is given here:

Definition1. Regression coefficient $R_B(A)$ of the event A with respect to the event B is called the difference between the conditional probability for the event A given the event B, and the conditional probability for the event A given the complementary event $\overline{B}$, namely

$R_B(A) = P(A \mid B) - P(A \mid \overline{B})$.

This measure of the dependence of the event A on the event B, is directed dependence.

The regression coefficient $R_A(B)$ of the event B with respect to the event A is defined analogously.

From the many interesting properties of the regression coefficients we would like to point out here just few:

(**R1**) The equality to zero $R_B(A) = R_A(B) = 0$ takes place if and only if the two events are independent.

(**R2**) The regression coefficients $R_B(A)$ and $R_A(B)$ are umbers with equal signs and this is the sign of their connection $\delta(A,B)$ =P(A∩B)-P(A)P(B). The relationships

$$R_B(A) = \frac{P(A \cap B) - P(A)P(B)}{P(B)[1-P(B)]},$$

and

$$R_A(B) = \frac{P(A \cap B) - P(A)P(B)}{P(A)[1-P(A)]}.$$

The numerical values of $R_B(A)$ and $R_A(B)$ may not always be equal. There **exists an asymmetry in the dependence between random events, and this reflects the nature of real life.**

(**R3**) The regression coefficients $R_B(A)$ and $R_A(B)$ are numbers between –1 and 1, i.e. they satisfy the inequalities

$$-1 \le R_B(A) \le 1; \qquad -1 \le R_A(B) \le 1.$$

(**R4.1**) The equality $R_B(A) = 1$ holds only when the random event A coincides with (or is equivalent to) the event B. Then is also valid the equality $R_A(B) = 1$;

(**R4.2**) The equality $R_B(A) = -1$ holds only when the random event A coincides with (or is equivalent to) the event $\overline{B}$ - the complement of the event B. Then is also valid $R_A(B) = -1$, and respectively $\overline{A} = B$.

We interpret the properties (r4) of the regression coefficients in the following way: As closer is the numerical value of $R_B(A)$ to 1, "as denser inside within each other are the events *A* and *B,* considered as sets of outcomes of the experiment". In a similar way we interpret also the negative values of the regression coefficient.

There is a symmetric measure of dependence between random events, and this is their coefficient of correlation.

Definition 2. Correlation coefficient between two events *A* and *B* we call the number

$$\rho_{A,B} = \pm\sqrt{R_B(A) \cdot R_A(B)},$$

where the sign, plus or minus, is the sign of the either of the two regression coefficients.

Remark. The correlation coefficient $\rho_{A,B}$ between the events A and B equals to the formal correlation coefficient ρ_{I_A,I_B} between the random variables I_A and I_B, the indicators of the two random events A and B.

The **correlation coefficient** $\rho_{A,B}$ between two random events is symmetric, is located between the numbers $R_B(A)$ and $R_A(B)$.

T**he following hold:**

ρ1. $\rho_{A,B} = 0$ holds if and only if the two events A and B are independent. The use of the numerical values of the correlation coefficient is similar to the use of the two regression coefficients. As closer is $\rho_{A,B}$ located to the zero, as "closer" to the independence are the two events A and B. For random variables similar statement is not true. The equality to zero of their mutual correlation coefficient does not mean independence

ρ2. The correlation coefficient $\rho_{A,B}$ always is a number between –1 and +1, i.e. $-1 \le \rho_{A,B} \le 1$.

ρ2.1. The equality $\rho_{A,B} = 1$ holds if and only if the events A and B are equivalent, i.e. when A = B.

ρ2.2. The equality $\rho_{A,B} = -1$ holds if and only if the events A and $\overline{B}$ are equivalent, i.e. when A = $\overline{B}$.

As closer is $\rho_{A,B}$ to the number 1, as "more dense one within the other" are the events A and B, and when $\rho_{A,B} = 1$, the two events coincide (are equivalent).

As closer is $\rho_{A,B}$ to the number -1, as "denser one within the other" are the events A and $\overline{B}$, and when $\rho_{A,B} = -1$, the two events coincide (are equivalent). Denser one within the other are then the events $\overline{A}$ and B.

2 The transfer rules

The above measures allow studying the behavior of interaction between any pair of numeric r.v.'s (X,Y) throughout the sample space, and better understanding and use of dependence.

Let the joint cumulative distribution function (c.d.f.) of the pair (X,Y) be $F(x,y)=P(X\le x, Y\le y)$, and marginals $F(x) =P(X \le x)$, $G(y)=P(Y\le y)$. Let introduce the events $A_x=\{x\le X\le x+\Delta_1 x\}$; $B_y=\{y\le Y\le y+\Delta_2 y\}$, for any $x, y \in (-\infty, \infty)$. Then the measures of dependence between events A_x and B_y turn into ***a measure of local dependence between the pair of r.v.'s X and Y on the rectangle*** $D=[x, x+\Delta_1 x]\times[y, y+\Delta_2 y]$. Naturally, they can be named and calculated as follows:

Regression coefficient of X with respect to Y, and of Y with respect to X on the rectangle $[x, x+\Delta_1 x]\times[y, y+\Delta_2 y]$. By the use of Definition 1 we get

$$R_Y((X,Y)\in D)= \frac{\Delta_D F(x,y)-[F(x+\Delta_1 x)-F(x)][G(y+\Delta_2 y)-G(y)]}{[F(x+\Delta_1 x)-F(x)]\{1-[F(x+\Delta_1 x)-F(x)]\}}.$$

Here $\Delta_D F(x,y)$ denotes the two dimensional finite difference for the function $F(x,y)$ on rectangle $D=[x, x+\Delta_1 x]\times[y, y+\Delta_2 y]$. Namely

$$\Delta_D F(x,y) = F(x+\Delta_1 x, y+\Delta_2 y) - F(x+\Delta_1 x, y) - F(x, y+\Delta_2 y) + F(x, y).$$

In an analogous way is defined $\rho_X((X,Y)\in D)$. Just denominator in the above expression is changed respectively.

Correlation coefficient $\rho_Y((X,Y)\in D)$ between the r.v.s X and Y on rectangle $D=[x, x+\Delta_1 x]\times[y, y+\Delta_2 y]$ can be presented in similar way by the use of Definition 2. We omit detailed expressions as something obvious.

It seems easier to find out the local dependence at a value $(X=i, Y=j)$ for a pair of discretely distributed r.v. (X,Y). Regression coefficient of X with respect to Y, and of Y with respect to X at a value $(X=i, Y=j)$ is determined by the rule

$$R_Y(X=i,Y=j)= \frac{P(X=i,Y=j)-P(X=i)P(Y=j)}{P(X=i)[1-P(X=i)]} = \frac{p(i,j)-p_{i.}p_{.j}}{p_{i.}(1-p_{i.})}.$$

Similarly, you can get that the local correlation coefficient between the values of the two r.v.'s (X,Y) is given by

$$\rho_{X,Y}(X=i,Y=j)= \frac{p(i,j)-p_{i.}p_{.j}}{\sqrt{p_{i.}(1-p_{i.})}\sqrt{p_{.j}(1-p_{.j})}}.$$

Using these rules one can see and visualize the local dependence between every pair of two r.v.'s with given joint distribution.

This ends our theoretical background of the local dependence structural study. Next we illustrate its application on qualitative and quantitative probability models.

3 Illustrations

3.1. Dependence in political interactions

In Esa and Dimitrov (2013a) it is shown that a multinomial model describes the entire spectrum of the party's life in the country, with N independent active *free i*ndividuals. The coordinates of the random vector $\vec{X}=(X_0, X_1, \ldots, X_r)$ represent the number of individuals members of each party $X_0 + X_1 + \ldots + X_r = N$. They are distributed according to the multinomial law

$$P(X_0=k_0, X_1=k_1, \ldots, X_r=k_r) = \frac{N!}{k_0!k_1!\ldots k_r!} P_0^{k_0} P_1^{k_1} \ldots P_r^{k_r},$$

$k_0 + k_1 + \ldots + k_r = N$.

The chance of the j^{th} party to exist (at a minimum M_j+1 *members* required for this purpose) is having Binomial probability $P(X_j \ge M_j+1)=1 - B(M_j; N, P_j)$ with the P_j, given by

$$P_0 = \left(1+\frac{\lambda_1}{\mu_1}+\cdots+\frac{\lambda_r}{\mu_r}\right)^{-1};$$

$$P_j = \frac{\lambda_j}{\mu_j}\left(1+\frac{\lambda_1}{\mu_1}+\cdots+\frac{\lambda_r}{\mu_r}\right)^{-1}, \quad j=1,2,\ldots,r.$$

$B(k; N, p)$ is notation for c.d.f. of Binomial distribution with parameters N and p, and k is its argument. Here λ_j and μ_j are the rates of joining and leaving party j correspondingly. P_o is the probability that a person stays independent on any party, and r is the number of active parties in the scene.

These results allow particular calculation of regression coefficients and correlation coefficients of the local dependence between any two components of the political life in the country. We get:

$$P(X_i=n,Xj=m)= \frac{N!}{n!m!(N-n-m)!}P_i^n P_j^m (1-P_i-P_j)^{N-n-m},$$

$$P(X_i=n)=\frac{N!}{n!(N-n)!}P_i^{N-n}(1-P_i)^{N-n};$$

$$P(X_j=m)=\frac{N!}{m!(N-m)!}P_j^{N-m}(1-P_j)^{N-m}.$$

Based on this we find that the regression local coefficients measuring the dependence between variables (X_i,X_j) at the point (n,m) are expressed by the equations

$$R_{X_j}(X_i=n,X_j=m) = \frac{\frac{(N-m)!}{m!(N-n-m)!}P_j^m(1-P_i-P_j)^{N-n-m}}{(1-P_i)^{N-n}\left(1-\frac{N!}{n!(N-n)!}P_i^n(1-P_i)^{N-n}\right)} -$$

$$\frac{N!P_j^m(1-P_j)^{N-m}}{n!(N-n)![1-\frac{N!}{n!(N-n)!}P_i^n(1-P_i)^{N-n}]}..$$

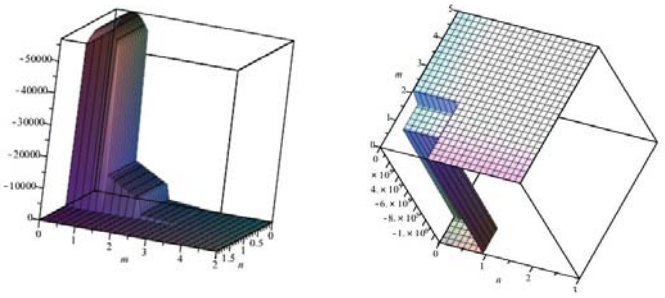
`By the way, this is the regression coefficient between any two components of a multinomial distribution.

Numeric example: Let us assume that in the main model we have r=4; $\lambda_1=\lambda_2=\lambda_2=\lambda_4=1$; $\mu_1=2$, $\mu_2=3$, $\mu_3=4$, and $\mu_4=5$. Then we will have:

$$P_0=\left(1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}\right)^{-1}=\frac{60}{137}.$$

Respectively we calculate $P_1=30/137$; $P_2=20/137$; $P_3=15/137$ and $P_4=12/137$.

With these numbers, and assuming for simplicity N=1000, we draw a surface of the local Regression coefficient surface of the dominating party 1 with respect to the next leading party 2. It looks as it is shown on the next figures. Next to it is the regression coefficient measure of dependence of the weakest party 4 on the strongest party 1. We see that they are negative when both parties have low results in votes, and that dependence is negligible when votes are higher. However, dependence of the weakest party goes low flat when it gets low number of votes.



3.2 Local dependence in reliability systems

In this case let us consider the two traditional systems in series and in parallel, of independent components, the system. We want to study how the regression coefficients of a component relbility varies in time with respect to the system reliability, and vise verse. For simplicity, consider system of just two components, since considering one component, everything else can be cobined as a second component. Results of these studies are shown next.

3.2.1. A system in series.

Assume, both components have live times exponentially distributed with parameters λ_1 and λ_2. Then the reliability function at any time instant t (this is the event B) equals $r(t)=e^{-(\lambda_1+\lambda_2)t}$, and the probability that component 1 functions (this is the event A) is $e^{-\lambda_1 t}$. The regression coefficient of the system with respect to component 1 is then

$$R_1(S)=\frac{r(t)-r(t)e^{-\lambda_1 t}}{e^{-\lambda_1 t}(1-e^{-\lambda_1 t})}=e^{-\lambda_2 t}.$$

Analogously we evaluate the regression coefficient of the component 1 with respect to the system at time t. It is given by the relation

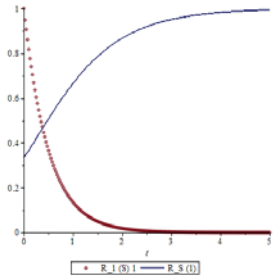
$$R_S(1)=\frac{r(t)-r(t)e^{-\lambda_1 t}}{r(t)[1-r(t)]}=\frac{1-e^{-\lambda_1 t}}{1-e^{-(\lambda_1+\lambda_2)t}}.$$

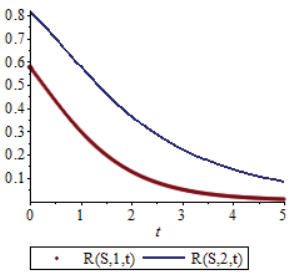
And the correlation coefficient between system reliability and the component reliability are changing during the time according to the relations

$$\rho_{s,1}(t)=\sqrt{\frac{e^{-\lambda_2 t}(1-e^{-\lambda_1 t})}{1-e^{-(\lambda_1+\lambda_2)t}}};$$

$$\rho_{s,2}(t)=\sqrt{\frac{e^{-\lambda_1 t}(1-e^{-\lambda_2 t})}{1-e^{-(\lambda_1+\lambda_2)t}}}.$$

Notice that all dependences are positive. Graphs of these functions of local dependence in time for λ_1=1 and λ_2=2 are shown on next figures.





We observe that the system reliability local correlation measures of dependence is decreasing to 0 for both components, but is higher with the weakest component 2, when the time increases. In the same time the regression coefficients between the system and the strongest component behave different: Local dependence $R_1(S)$ approaches 0 with the time (like system becomes independent on component 1with the growth of the time) when the local dependence $R_S(1)$ of strongest component 1 on the system reliability approaches 1 with the growth of the time.

3.2.2. System in parallel.

Assume again, both components have exponentially distributed live times with parameters λ_1 and λ_2. Then the reliability function at any time instant t (this is the event B) equals

$$r(t)=1-(1-e^{-\lambda_1 t})(1-e^{-\lambda_2 t}),$$

and the probability that component 1 functions (this is the event *A*) is $e^{-\lambda_1 t}$. Applying the rules we obtain:

The regression coefficient of the system with respect to component 1 is then

$$R_1(S)=\frac{r(t)-r(t)e^{-\lambda_1 t}}{e^{-\lambda_1 t}(1-e^{-\lambda_1 t})}=1-e^{-\lambda_2 t}.$$

Analogously we evaluate the regression coefficient of the component 1 with respect to the system at time t. It is given by the relation

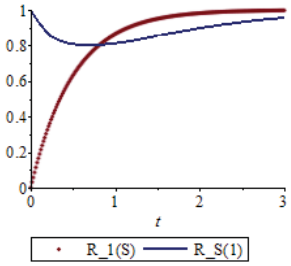
$$R_S(1)=\frac{r(t)-r(t)e^{-\lambda_1 t}}{r(t)[1-r(t)]}=\frac{e^{-\lambda_1 t}}{1-(1-e^{-\lambda_1 t})(1-e^{-\lambda_2 t})}.$$

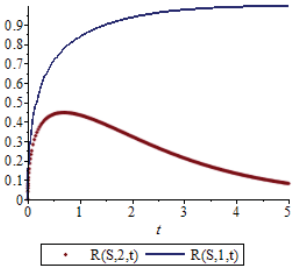
And the correlation coefficient between system reliability and the component reliability are changing during the time according to the relations

$$\rho_{s,1}(t)=\sqrt{\frac{e^{-\lambda_1 t}(1-e^{-\lambda_2 t})}{1-(1-e^{-\lambda_1 t})(1-e^{-\lambda_2 t})}};$$

$$\rho_{s,2}(t)=\sqrt{\frac{e^{-\lambda_2 t}(1-e^{-\lambda_1 t})}{1-(1-e^{-\lambda_1 t})(1-e^{-\lambda_2 t})}}.$$

Notice that all dependences are positive. Graphs of these functions of local dependence in time for λ_1=1 and λ_2=2 are shown on next figures. First graph represents the two regression coefficients (strongest component reliability vs system reliability, and system reliability vs same component) superimposed on the same graph. Asymetric dependence is obvious fact. The second graph represents the two cccomponents-system correlation coefficients. We see that the system reliability local correlation measure of dependence is approaching 1 with the strongest component 1, and approaches 0 with the weakest component when the time increases. In the same time the both regression coefficient between the system and the strongest component approach 1 with the growth of the time.





3.3. Categorical variables

The most interesting and valuable applications in the Big Data analysis we see in the analysis of local dependences between non-numeric vs non numeric variables, as well as between non-numeric vs numeric variables. Since analysis in this kind of studies is (according to us obviously) too similar, we give here as an example of local dependence between categories of two non-numeric random variables. It is just an illustration of the proposed measures of dependence between random events. We analyze here an example from the book of Alan Agresti [1]. The following table represents the observed data about the yearly income of people and the job satisfaction.

Table1. Observed Frequencies of Income and Job Satisfaction

	Job Satisfaction			
Income US $$	**Very Dissatisfied**	**Little Satisfied**	**Moderately Satisfied**	**Very Satisfied**
< 6,000	**20**	**24**	**80**	**82**
6,000–15,000	**22**	**38**	**104**	**125**
15,000-25,000	**13**	**28**	**81**	**113**
> 25,000	**7**	**18**	**54**	**92**
Total Marginally	**62**	**108**	**319**	**412**

The probabilities in each category

$$P_{i,j} = \frac{n_{i,j}}{n}, P_{i,.} = \frac{n_{.i}}{n}, P_{.,j} = \frac{n_{.j}}{n},$$

the above table produce the join empirical distribution of the two categories. $P_{i,j}$ is the probability that a new observation will fall in the respective subcategory i by income, and subcategory j by job satisfaction. Table 2 presents the join distribution for the two categorical variables.

Table 2: Join probability distribution and marginal distributions (Income, Job Satisfaction) $P_{i,j}, P_{i..}, P_{.j}$

Job Satisfaction					
Income US $$	**Very Dissatisfied**	**Little Satisfied**	**Moderately Satisfied**	**Very Satisfied**	**Total (marginal) distribution**
< 6,000	**.02220**	**.02664**	**.08879**	**.09101**	**.22864**
6,000–15,000	**.02442**	**.04217**	**.11543**	**.13873**	**.32075**
15,000-25,000	**.01443**	**.03108**	**.08990**	**.12542**	**.26083**
> 25,000	**.00776**	**.01998**	**.05993**	**.10211**	**.18978**
Total) distr.	**.06881**	**.11987**	**.35405**	**.45727**	**1.00000**

Applying the rules for the proposed measures of dependence between random events, and using the empirical probabilities in Table 2, we obtain these measures as shown in the tables from 3 to 5. Positive sign indicates a positive local dependence between the two sub-categories, and the negative sign indicates the opposite in this locality. The negative association is marked in light blue, and the positive areas of association are highlighted in yellow.

Since numbers speak less than graphs, we immediately give a graphic presentation of these two-argument (cross sections of any two sub-categories) functions. The ancient Greeks use to say „ Just watch and conclude".

Table 3. Empirical Estimations of the regression coefficient between each particular level of income with respect to the job satisfaction $r_{Satisfaction_j}(IncomeGroup_i)$

Job Satisfaction				
Income US $$	**Very Dissatisfied**	**Little Satisfied**	**Moderately Satisfied**	**Very Satisfied**
< 6,000	0.1009327	-0.00727	0.034281	-0.05456
6,000–15,000	0.0366630	0.035276	0.00817	-0.03199
15,000-25,000	-0.054899	-0.00176	-0.0107	0.024782
>25,000	-0.082696	-0.02625	-0.03175	0.061768

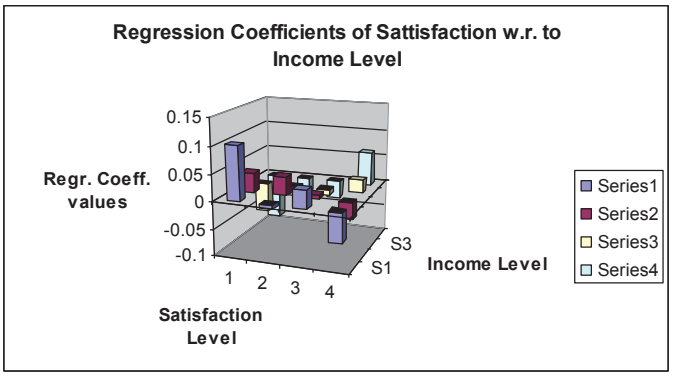
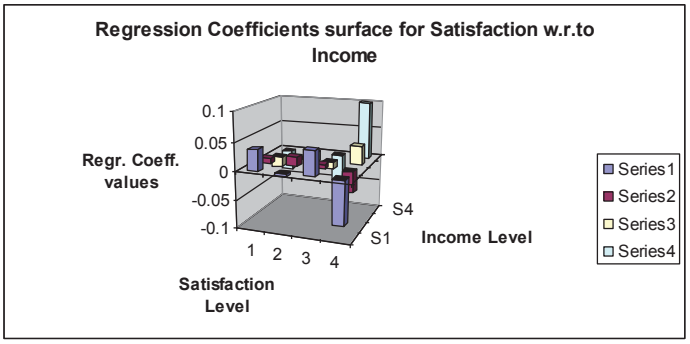


Table 4: Empirical Estimations of the regression coefficient between each particular level of the job satisfaction with respect to the income

Job Satisfaction				
Income US $$	Very Dissatisfied	Little Satisfied	Moderately Satisfied	Very Satisfied
< 6,000	0.0366701	-0.00435	0.044454	-0.07677
6,000–15,000	0.0107825	0.017082	0.008576	-0.03644
15,000-25,000	-0.018245	-0.00096	-0.01269	0.0319
> 25,000	-0.034460	-0.01801	-0.04723	0.099694



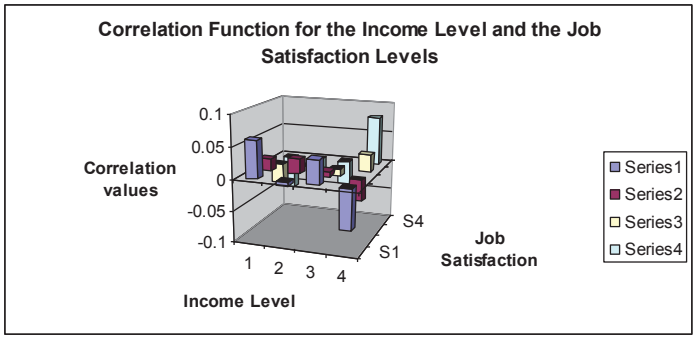
For instance, the number $r_{VeryDissatisfied}(< 6000) = .100932704$ indicates positive dependence of the category of the lowest income "<6,000" on the category: "Very Dissatisfied" for the Job Satisfaction. The same number with negative sign $r_{VeryDissatisfied}(\overline{< 6,000}) = -.100932704$ indicate the negative strength of dependence of all the other income categories, higher than "<6,000" on the category: "Very Dissatisfied" for the Job Satisfaction. Similarly to the connection function, the sums of numbers from several cells in a column of Table 3, (or in a row of Table 4) will indicate the strength of dependence of the union of the sub-categories of the respective factor "Income" on the corresponding to the column sub-category of the "Job,

Satisfaction" (with analogous switch of factor's interpretation).

The two regression coefficient' matrices allow calculating the correlation coefficients between every pair of sub-categories of the two factors. Table 5 summarizes these calculations. The numbers actually represent the numerical estimations of the respective mutual local correlation coefficients. Obviously, each of these numbers gives the **local average measure of dependence** between the two factors. Unfortunately, the summation of the numbers in a vertical or horizontal line is not having the same or similar meaning as in the cases of regression in regression matrices. Also, the sums of the numbers in a row or in a column do not equal zero as above.

Table 5: Empirical Estimations of the correlation coefficient between each particular income group and the categories of the job satisfaction ρ(*Income Group, Job Satisfaction*)

	Job Satisfaction			
Very Dissatisfied	Very Dissatisfied	Little Satisfied	Moderately Satisfied	Very Satisfied
< 6,000	0.060838	- 0.005623	0.039037	- 0.064721
6,000–15,000	0.019883	0.024548	0.008371	- 0.034144
15,000-25,000	- 0.031649	- 0.001302	- 0.011653	0.028117
> 25,000	- 0.053383	- 0.02174	- 0.038723	0.078472



4 Conclusions

We extend our previous study of local dependence between random events to measures of local dependences between random variables. This turns into a study of the local dependence on a rectangle, where interval values of the random variables meet. These local dependences are universally valid and possibly can be continued for higher dimensions. As illustrations we consider local dependences in politics and reliability systems. The numerical illustrations can be graphically visualized. Visualizations show that local dependence is essentially different on different areas in the field. Graphics offer much more comments and further thoughts Our expectations are that the analysis of Big Data sets will be enriched with the inclusion of our approach into its system tools.

5 Reference

[1] A. Agresti . Categorical Data Analysis, New York, John Wiley & Sons, 2006.

[2] Dimitrov. Measures of dependence in reliability, Proceedings of the 8-th MMR'2013, Stellenbosh, South Africa, July 1-4 , pp 65-69, 2013.

[3] B. Dimitrov. Some Obreshkov Measures of Dependence and Their Use Compte Rendus de l'Academie Bulgare des Sciences, V. 63, No.1, pp. 15-18, 2010.

[4] S. Esa and B. Dimitrov. Dependencies in the world of politics, Proceedings of the 8-th MMR'2013, Stellenbosh, South Africa, July 1-4 , pp 70 – 73, 2013.

[5] S. Esa and B. Dimitrov "Survival Models in Some Political Processes", Risk Analysis and Applications (RT&A), v. 8, # 3, (30) pp 97 – 102, 2013.

Improving the Feature Selection for the Development of Linear Model for Discovery of HIV-1 Integrase Inhibitors

Matineh Kashani Moghaddam, Richard Adrian Galvan, and Ahmad Reza Hadaegh
Department of Computer Science , California State University, San Marcos, California, USA

Abstract - *Dimeric aryl β- diketo acids have proven to be effective inhibitors to the HIV strand transfer mechanism of HIV-integrase. In order to create the best drug to fight HIV-integrase, it is important to know which features of the diketo acids have the biggest impact on reducing HIV enzyme activity. In this research, the development of the Differential Evolutionary Binary Particle Swarm Optimization (DE-BPSO) algorithm with a Multiple Linear Regression (DE-BPSO/MLR) model is discussed and compared with the results against linear models tested in previous research. The use of both of evolutionary algorithms and predictive models, such as the differential evolutionary – binary particle swarm optimization (DE-BPSO) algorithm can help find a subset of the diketo acid's chemical descriptors that are best able to predict the reduction in HIV-integrase enzyme activity by more than 50%.*

Keywords: Drug Design, HIV-1 Integrase inhibitors, Evolutionary algorithms, DE-BPSO, Data-mining

1 Introduction

The human immunodeficiency virus type 1 (HIV-l) is a retrovirus responsible for the acquired immunodeficiency syndrome (AIDS) disease. According to Global Health Observatory Data in the year of 2013 almost 78 million people have been infected with the HIV virus and around 39 million people have died of HIV [1].

One of the biggest problems with current HIV drugs on the market is their lack of resistance to the mutation of HIV proteins. There are three HIV enzymes: HIV-Protease, Integrase, and Reverse Transcriptase that represent the core replication process of this disease. The majority of the drugs on the market target the HIV-Protease and Reverse Transcriptase enzymes. However, mutations arising in the HIV-l genome which confer resistances to existing anti HIV-1 inhibitors drive the need to develop new anti-HIV-1 drugs with an acceptable mutation profile. In order to develop a stronger HIV mutant resistant drug, it is important to know which physiochemical properties of the drug will lead to the biggest reduction in HIV enzyme activity. This research focuses on predicting inhibitors to target HIV-Integrase [2,3,4].

Quantitative structure-activity relationship (QSAR) models and evolutionary algorithms can be used in combination to determine which chemical descriptors of a drug will lead to the biggest reduction in enzyme activity. Given a set of quantifiable features and an associated target variable, QSAR models determine the relationship each feature has with the target variable [5]. Knowing which chemical feature descriptors have the biggest impact on enzyme activity helps drug manufacturers focus on the descriptors that will produce the best HIV mutant resistant drug. Through the use of QSAR models and evolutionary algorithms, we can create a model with the best set of features to predict the effect a drug has on enzyme activity and thus how well the drug should be able to overcome the mutation of HIV proteins.

In this research, we explore alternative approaches to the Multiple Linear Regression methods used in previous research. The focus is to apply commercial MLR modelling methods to find descriptors that can reduce the activity of HIV-Integrase enzyme activity protein by at least 50%. The work is executed on an experimental sample of 91 aryl beta-diketo acids. The result is then compared with the result of the best linear MLR model trained on a series of 37 aryl beta-diketo acids as part of previous investigative research [6, 7].

2 Background

2.1 Quantitative structure-activity relationship and computational chemistry

Quantitative structure-activity relationship (QSAR) modeling is a paradigm used to support virtual drug screening [8, 9]. QSAR is used to find relationships between the variations in the values of molecular properties and the biological activity for a series of compounds so that these "rules" can be used to evaluate new chemical entities. QSAR is a mathematical relationship between biological activity of molecular system and its geometric and chemical characteristics:

Activity = f (molecular or fragmental properties)

Therefore, QSAR is used to understand drug action, design new compounds, and screen chemical libraries and works on

the assumption that structurally similar chemical descriptors have similar activities.

2.2 Models and methods used in this thesis

In this thesis, we use a commercial implementation of linear model Multiple Linear Regression (MLR) to develop a model to identify a subset of the Aryl β-diketo acid chemical descriptors that are best able to predict the reduction in HIV-integrase enzyme activity. We run DE-BPSO using our model, then compare our results to a comparable model used in previous research. Comparing models will help determine which QSAR model yields the most effective HIV-1 inhibitors.

3 Methods

3.1 Dataset and descriptor calculation

In this research, a dataset of 91 dimeric aryl β-diketo acids which has been shown experimentally to inhibit the strand transfer function of HIV-1 integrase have been studied. The biological activity of each molecule was reported in the form of pIC50, a measure of the concentration of a drug needed to inhibit the biological activity of the target HIV-1 enzyme by 50%.

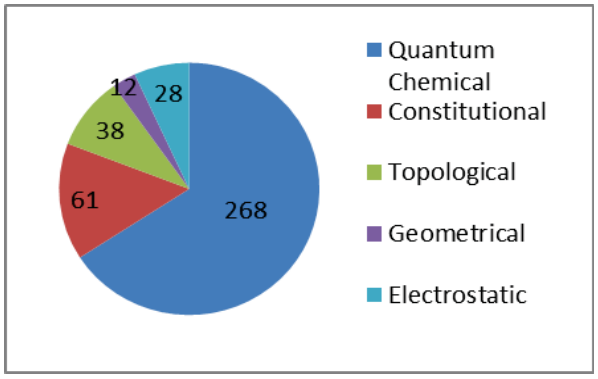


Figure1. Descriptors (constitutional, geometrical, topological, electrostatic, and quantum-chemical)

Each structure was geometrically optimized at the RM1 (RM1) level using AMPAC (AMPAC) [10] followed by chemical descriptor calculation using CODESSA (Codessa) [11].

A total of 385 constitutional, geometrical, topological, electrostatic, quantum-chemical descriptors were considered for the study (Figure 1). All descriptor values were rescaled to have a zero mean and a standard deviation of one. The dataset is divided into a training (n=55), validation (n=18), and test (n=18) set using the Kennard Stone algorithm [12]. This ensures maximization of the chemical space during model training and accurate tests within the space.

3.2 Binary Particle Swarm Optimization

Particle swarm optimization (PSO) simulates the behavior of flocks of birds [13]. In this scheme, each single solution is modeled as a particle. Every particle has a fitness (cost) value that is evaluated by the corresponding fitness function. In addition, each particle has a velocity that adjusts its direction toward a global optimal solution.

In PSO, we start with a group of particles that make one population; with each particle modeling one solution. The particles fly through the problem space by following the current optimum particles. In every iteration, each particle is updated by following two "best" values. The first one is the best solution (fitness) that it has achieved so far. This value is called local best ($\mathbf{l}_{\mathbf{best}}$).Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is the global best among the swarm and is called ($\mathbf{g}_{\mathbf{best}}$).In addition, each particle has a current velocity, which represents a magnitude and direction toward a new, presumably better solution modeled as an optimal position in the exploration space. A particle also has a measure of the quality of its current position, the particle's best known position (that is, a previous position with the best known quality), and the quality of the best known position.

The Differential-Evolution (DE) algorithm is a population based algorithm. It relies on mutation and selection operations to optimize its solutions. The algorithm uses mutation operation as a search mechanism and selection operation to direct the search toward the prospective regions in the search space. The main steps of the DE algorithm are given as Initialization, Evaluation, (Repeat: Mutation, Recombination, Evaluation, Selection, until (termination criteria are met).

3.3 Differential Evolution – Binary Swarm optimization algorithm (DE-BPSO)

Recently, a hybrid differential evolution-binary particle swarm optimization (DE-BPSO) algorithm has been proposed [6]. This algorithm can be used for feature selection to create the descriptor subsets which are then used in the development of quantitative structure-activity relationship (QSAR) models:

1) First the BPSO is used as a feature selection algorithm. In the initial generation, the velocity and position vector of BPSO are initialized according to:

$$v_{id} = \text{rand}(0,1)$$

$$x_{id} = \begin{cases} 1 \rightarrow \text{If } v_{id} <= \lambda \\ 0 \rightarrow \text{otherwise} \end{cases} \quad (1)$$

where λ is the probability of selecting a descriptor. To keep the total number of selected descriptors low for faster calculation for each particle we set λ= 0.015 (1.5%). To overcome local optima convergence issues, Shen et al. [14] had 20% of the particles search the space randomly. Instead of a random particle search, in DE-BPSO the set of rules have been modified to incorporate a positional bit mutation factor:

$$x_{id} = \begin{cases} x_{id}, \ if \ 0 < v_{id} < \alpha \\ p_{id}, \ if \ \alpha < v_{id} \leq \frac{1}{2}(1+\alpha) \\ p_{gd}, \ if \ \frac{1}{2}(1+\alpha) < v_{id} \leq (1-\beta) \\ 1 - x_{id}, \ if \ (1-\beta) < v_{id} \leq 1 \end{cases} \quad (2)$$

The α value in this research changes from 0.5 and decreases to 0.33 as we go from one generation to another. The higher value of α drives the direction of particle toward the global best positon, while the lower value of α drives the particle toward the local best position in the swarm.

β is the percentage chance of flipping a bit. The β value and the randomized selection of 20% of the new population are used to overcome local optima convergence issues.

2) Next, the velocities of the particle swarm are evolved using DE to obtain the position vector using the above rules. These rules are applied to compute the velocity vector vi of each individual particle at each iteration:

$$v_i' = v_a + F(v_b - v_c) \quad (3)$$

$$v_{i_d} = \begin{cases} v'_{i_d} & rand(0,1) < CR \\ v_{i_d} & otherwise \end{cases} \quad (4)$$

The values of the position vector vi are then determined according to the set of rules in Eq. (1). In this research, *F* controls the length of the exploration vector from velocity $v_{a.}$ and CR is the crossover rate, representing the probability of an individual's feature descriptor, as modeled by dimension *d*, being exchanged with the provisional offspring's feature descriptor in the corresponding dimension. For our research *F* = 0.7 and *CR* = 0.8 as it was used in the previous work [6].

Previous studies have shown that developing a hybridized EA based feature selection method for developing QSAR models using DE and BPSO is an improvement over the search performance of standalone BPSO algorithm. To determine the optimal parameters for the BPSO and DE-BPSO feature selection algorithms on this dataset, 500 simulations were run for each of the varying parameters and averaged the fitness over 2000 generations.

The DE-BPSO algorithm was then used to develop multiple linear regression (MLR) models for the analysis of aryl β-diketo acid compounds for the inhibition of HIV-1 integrase. This algorithm is used as the feature selection method to select the features with the most significant influence on the compounds for the inhibition of HIV-1 integrase.

In this research, we are working with a set of 91 aryl β-diketo acid compounds which are more than twice the number of experimental compounds used in previous research [6, 7].

3.4 The machine learning model and data-mining tool used in this research

In this research, DE-BPSO is used as the feature subset algorithm which serves as input to the linear model. The linear-based QSAR model trained using Multiple Linear Regression and coupled with DE-BPSO algorithm is improved and the results have been compared with the work previously done with smaller set of experimental data [6, 7]. Some of the results that we discuss include the value of R^2 (coefficient of variation) and MSE (mean-squared error) which are computed and compared to corresponding results produced by linear QSAR model (MLR) from the previous research. These values aid in determining which models provide more specific and effective inhibitory drug proposals.

4 Results and Discussion

4.1 The result of developing linear QSAR model

In this research DE-BPSO algorithm is used to select the individual descriptors. Machine learning models are then developed from each of the selected descriptor sets selected by each individual particle. The fitness ($cost$) of the selected descriptors is measured by the RMSE, sample size (m) of both the training (t) and validation (v) set, number of descriptors in the model (n), and a parsimonious penalty factor γ.

The value of γ is used to maintain the balance between finding models with over-fit and under-fit and would be monitored to find a value suitable for a given dataset. To obtain predictive QSAR models for this dataset we found the appropriate value of γ by trial and error as γ = 3.3. This value of gamma is the optimal experimental value derived by previous research on DE-BPSO. To evaluate the MLR models generated from the descriptors selected by the DE-BPSO, we used the fitness function below [6]. The fitness function is

designed to minimize the number of descriptors in the model and prevent over-fitting.

$$f = \sqrt{\frac{(m_t - n - 1) \cdot (RMSE_t^2 + m_v \cdot RMSE_v^2}{m_t - \gamma \cdot n - 1 + m_v}} \qquad (5)$$

Where m_t and m_v are the sample size of training set and validation set respectively, and RMSEs refer to the root mean square error of training and validation sets. The top ranked models with the lowest cost are analyzed and interpreted to understand the physiochemical properties of dimeric Aryl β-Diketo acids most conducive toward biological inhibition of HIV-1. To develop models for the analysis of aryl β-diketo acids, we used the optimal parameters found in simulations of previous research (β= 0.004, F = 0.7, CR = 0.8) **[6]** with a population size of 50 individuals in the particle swarm and 1000 generations for the DE-BPSO feature selection algorithm.

Each model is evaluated in terms of the coefficient of determination, R^2, mean squared error (MSE) and root-mean squared error (RMSE). In order to compute R^2 for all models, the following definition is considered, where by $\bar{y}$ denotes the target's mean value:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y}_i)^2}. \qquad (6)$$

It should be noted that this version of R^2 can be negative if predictions are worse than forecasting the mean. In particular, R^2 is considered as a singular measure of predictive accuracy because it represents the amount of variation of the dependent variable explained by the model.

MLR models with high correlation (R^2 > 0.6), high predictive correlation with the validation (R_v^2 > 0.5) and test (R^2_{test} > 0.5) sets, and cross-validated quality of fit (Q^2 > 0.5) are considered for analysis.

The model with the lowest cost function has test set statistics R^2_{train}= 0.8967. The model has 5 descriptors:

Table1. Model Performance (DE-BPSO-MLR)-(91 β-diketo acids)

$RMSE_t$	R^2_t	Q^2_t	$RMSE_V$	$R2_V$	$RMSE_{test}$	R^2_{test}
0.3729	0.8967	0.6614	0.1210	0.8423	0.0953	0.8098

Table2. Selected Descriptors in β-Diketo acid QSAR Model-91 β -diketo acids (DE-BPSO-MLR)

Variable	Coefficient	Descriptor
X1	0.272	Relative number of aromatic bonds
X2	0.789	Max atomic state energy for a H atom
X3	0.452	Relative number of C atoms
X4	0.558	Relative number of single bonds
X5	0.284	ESP-RNCG Relative negative charge (QMNEG/QTMINUS)

Based on this optimal QSAR model, the feature descriptors with the greatest influence on the biological activity of HIV-1 integrase are the Relative number of single bonds and Relative number of C atoms. The positive coefficient of both descriptors indicates that a more hydrophobic inhibitor with a greater partial positive charge is conducive for inhibition.

5 Conclusion and future work

This research is done on a data set of 91 dimeric aryl β -diketo acids (Train set =55, Validation set = 18 and Test set = 18) using Linear Model (MLR) and DE-BPSO algorithm as the feature selection method. Since, the DE-BPSO is totally a new hybrid algorithm that has been introduced recently; no significant work had been done previously using this algorithm. The only earlier research available to compare has been on 37 dimeric aryl B-diketo acids (Train set =23, Validation set = 7 and Test set = 7) using Linear Model (MLR) and DE-BPSO algorithm as the feature selection method **[6]**. So, we are comparing our result from that research with our result.

6 References

[1]. http://www.who.int/gho/hiv/en/

[2]. Smith RA, Raugi DN, Pan C, Sow PS, Seydi M, Mullins JI, Gottlieb GS. In vitro activity of dolutegravir against wild-type and integrase inhibitor-resistant HIV-2. University of Washington-Dakar, HIV-2 Study Group. Retrovirology. 2015 Feb 5;12(1):10. doi: 10.1186/s12977-015-0146-8 PMID: 25808007

[3]. ang H, Kowalski MD, Lakdawala AS, Vogt FG, Wu L. An efficient and highly diastereoselective synthesis of GSK1265744, a potent HIV integrase inhibitor. Org Lett. 2015 Feb 6;17(3):564-7. doi: 10.1021/ol503580t. Epub 2015 Jan 23. PMID: 25615910

[4]. Radzio J, Spreen W, Yueh YL, Mitchell J, Jenkins L, García-Lerma JG, Heneine W. The long-acting integrase inhibitor GSK744 protects macaques from repeated intravaginal SHIV challenge. Sci Transl Med. 2015 Jan 14;7(270):270ra5. doi: 10.1126/scitranslmed.3010297. PMID: 25589631

[5]. Csermely, K. Structure and Dynamics of Molecular Networks: A Novel Paradigm of Drug Discovery. Retrieved from www.ncbit.nlm.nih.gov 2013.

[6]. Gene M. Ko, Rajni Garg, Sunil Kumar, Barbara A. Bailey, Ahmad R. Hadaegh, "Differential Evolution-Binary Particle Swarm Optimization for the Analysis of Arylbeta-diketo Acids for HIV-1 Integres Inhibition". WCCI 2012 IEEE World Congress on Computational Intelligence June, 2012 Brisbane Australia, pp. 1849-1855

[7]. Gene M. Ko, A. Srinivas Reddy, Rajni Garg, Sunil Kumar, Ahmad R. Hadaegh, "Computational Modeling Methods for QSAR Studies on HIV-1 Integrase Inhibitors (2005-2010)," Current-Computer-Aided Drug Design, **2012**, pp. 255-270.

[8]. Moonsamy S, Dash RC, Soliman ME. Integrated computational tools for identification of CCR5 antagonists as potential HIV-1 entry inhibitors: homology modeling, virtual screening, molecular dynamics simulations and 3D QSAR analysis. Molecules. 2014 Apr 23;19(4):5243-65. doi: 10.3390 / molecules 19045243. PMID: 24762964

[9]. Kumar S, Tiwari M. Variable selection based QSAR modeling on Bisphenylbenzimidazole as Inhibitor of HIV-1 reverse transcriptase. Med Chem. 2013 Nov;9(7):955-67. PMID: 23106285

[10]. AMPAC version 9.2. 1. Shawnee, Kansas: Semichem, Inc., 2009.

[11]. CODESSA version 2.7.16. Shawnee, Kansas: Semichem, Inc., 2009

[12]. R.W.Kennard, L.. Fundamentals of Computational Swarm Intelligence. 1969.

[13]. Kennedy, J. Review of Elgelbrecht's Fundaentals of Computational Swarm Intelligence. Genet Program Evolvable. March (2007) 8:107-109 DOI 10.1007/s 10710-006-9020-8.

[14]. Shen, J. Jiang, C. Jiao, G. Shen, and R. Yu. "Modified Particle Swarm Optimization Algorithm for Variable Selection in MLR and PLS Modeling: QSAR Studies of Antagonism of Angiotensin II Antagonists". European Journal of Pharmaceutical Sciences, vol 22, issue 2-3, pp. 145-152,2004

Block-matching Twitter data for traffic analysis

A. Shuqair, and S. P. Kozaitis
Department of Electrical and Computer Engineering, Florida Institute of Technology, Melbourne, FL, USA

Abstract – *The successful design of a social media based system for traffic analysis in emergencies depends on the ability to adapt to differences in speech. We developed a system that is data-driven and relies mostly on patterns of tagged speech to identify the conditions of roads from Twitter data. We analyzed blocks of tweets that are similarly tagged to arrive at a decision about the condition of road. The system is continuously updated so real-time operation is possible.*

Keywords:, *block matching, parts-of-speech, social media, Twitter*

1 Introduction

Natural disasters occur frequently and affect many thousands of people and properties each year. The most notable are volcanoes, earthquakes, floods and hurricanes, along with other types that occur less often. In the United States, the incidence of hurricanes is a major concern, such as in the State of Florida due to its location and formation as a peninsula with the surrounding waters of the Atlantic Ocean and Gulf of Mexico. Hurricanes have many hazards associated with them, including, heavy rainfall, extreme wind gusts, and flooding, which can potentially cost billions of dollars in damages. It is not surprising that hurricane season in Florida has been associated with major traffic problems when large numbers of people evacuate from the coastal regions. Solutions are needed to help guide drivers safely and efficiently during these evacuations in times of potential crisis.

Our objective was to use advanced signal processing methods applied to text streams to help drivers avoid traffic congestion throughout the hurricane season. Ideally, our approach will predict current traffic status, thereby allowing effective planning and providing satisfactory traffic information. The proposed system will detect current trending events from Twitter streams, analyze the tweets, and then provide a probabilistic method toward arranging a secure and swift evacuation route.

Twitter is considered to be one of the most popular global social networks. It allows users to upload data within a range of 140 characters. Twitter first launched in March 2006, and since that time there are approximately 255 million monthly active users and 500 million Tweets sent per day. [1] Considering this, Twitter may potentially be part of a solution when used as a communication system during natural disasters or other unforeseen crises.

There have been several attempts to use Twitter and other social media to provide real-time traffic information [2-5] with text mining or natural language processing (NLP) often being used. In one system, words were tokenized, matched to a database and classified into one of 8 categories of words such as adjective, noun, verb, etc. [2]. Next, sentence analysis was carried out using rules to gather information on time, origin, destination and traffic condition. The data was filtered with a template to extract relevant data for visualization on a map with conditions rated on a scale from 1 – 3. It was noted that about 50% of the tweets had words in the vocabulary or didn't fit the rules.

Another method considered a limited number of senders that were primarily official in capacity. [3] After tweets were tokenized, the tokens were classified into 12 categories that were either considered events or situations. Then, an exact string matching process was used to find street names in a large database, and fuzzy string searching using gazetteers was performed to match the street by crossroads and neighborhood names. The main concern was how to geocode and locate the tweets. They used Twitter users as sensors and created their own dictionary from gazetteers called GEODEC that contained all the abbreviations of the location data. After geocoding the traffic data they displayed it for users in a map view.

Another method used an existing tokenizer called Lexto to analyze and classify Twitter data by keywords, which described traffic conditions using the Twitter search API [4]. This approach classified road data either into a point(e.g. an intersection) or a link (e.g. a road) then into 8 subcategories such as place, verb, etc. Tweets were limited to traffic keywords such as *accident* and *traffic congestion,* with a large dictionary created for each category. Rules were then followed to refine the tweets for traffic conditions, and tokens were further analyzed to determine start and end points and other parameters.

In our proposed system, we examined Twitter data in terms of patterns of tagged speech rather than using keywords or specific tags as in most other approaches. We looked for patterns of text and grouped like patterns together in what we refer to as a block-matching approach that was also data-driven. We examined groups of like patterns to extract information such as the condition of a road. This approach has the advantage that it can adapt to different syntaxes and potentially different languages.

2 System

Given the importance of social media networks in the analysis and extraction of traffic data, we considered Twitter data streams as the primary source of our data. However, the nature of that data required us to construct a plan for its extraction, classification, and analysis. When data is collected, tweets must first be determined if they are related to traffic issues. After unnecessary symbols are removed, each tweet is tokenized and each token is tagged as a part of speech (POS) such as an adjective, noun etc. As more data is collected, similar sequences of tags are placed in groups with the same tag pattern. For example, a group may be formed by a collection of parts of a tweet that have a pattern of *noun/participle/adjective*, and other groups formed depending on the pattern of the tags. We used a text parsing POS method using a Stanford parser [6] and Carnegie Mellon parser [7]. A sample set of common syntax categories are listed in the table below.

CC	Coordinating conjunction	TO	to
CD	Cardinal number	UH	Interjection
DT	Determiner	VB	Verb, base form
EX	Existential there	VBD	Verb, past tense
FW	Foreign word	VBG	Verb, gerund/present participle
IN	Preposition/subordinating conjunction	VBN	Verb, past participle
JJ	Adjective	VBP	Verb, non-3rd person singular present
JJR	Adjective, comparative	VBZ	Verb, 3rd ps. sing. present
JJS	Adjective, superlative	WDT	Wh-determiner
LS	List item marker	WP	Wh-pronoun
MD	Modal	WP	Possessive wh-pronoun
NN	Noun, singular or mass	WRB	Wh-adverb
NNS	Noun, plural	#	Pound sign
NNP	Proper noun, singular	$	Dollar sign
NNPS	Proper noun, plural	.	Sentence-final punctuation
PDT	Predeterminer	,	Comma
POS	Possessive ending	:	Colon, semi-colon
PRP	Personal pronoun	(	Left bracket character
PP	Possessive pronoun	)	Right bracket character
RB	Adverb, comparative	"	Straight double quote
RBR	Adverb	`	Left open single quote
RBS	Adverb, superlative	"	Left open double quote
RP	Particle	'	Right close single quote
SYM	Symbol	"	Right close double quote

Table 1 Examples of POS tags used.

A set of rules is used to keep only potentially useful patterns of tags referred to here as blocks. As more tweets are considered, similar collections of blocks, called groups are created before further analysis. In our case each group is analyzed further to determine if its content refers to a road, what road, and whether the road is passible. This is accomplished by examining the entire group rather than just a particular block. Once the necessary information has been extracted it is stored in a database for display on a map and the process continues in the following steps.

1 Gather and preprocess tweets

2 Eliminate non-traffic tweets

3 Tag tweets

4 Group tagged tweet (block) with similar blocks

5 Repeat cycle for some number of tweets

6 Analyze groups of blocks and eliminate unnecessary blocks

7 Combine or split blocks

8 Extract pertinent information of blocks

2.1 Acquiring Tweets

In this step the system used Twitter APIs (stream API + REST API) for getting the data from Twitter streams. The APIs were used to search for predefined keywords (e.g. street names, events) in a specific area or city to retrieve a set of real-time tweets. For the client, we are using the internet and for the search keywords we sent an HTTP request from a server that calls the Twitter REST API for previous tweets. The Twitter Streaming API was used for real-time tweets.

When a user types a search keyword from the interface (presentation layer) it calls the request processor class then the business layer which will retrieve the previous tweets from the Twitter REST API, and live tweets from the Twitter Streaming API. Finally, the processed data will be ready for storage in the database. The first step after fetching the tweets from Twitter streams is a preprocessing step, because it removes unwanted symbols from tweets such as emoticons, symbols, retweets and URL's.

2.2 Classifying traffic tweets

The preprocessed tweets were used in this step as an input, and a process of classifying traffic-related events follows in the next step. By using a gazetteer the system will check each tweet and compare it with entries looking for street names. If a street name is detected then the tweet will go to the next tweet otherwise it will be disregarded.

2.3 Grouping

At this level of the process the traffic tweets are ready to be examined and classified into a number of groups. The system follows a data-driven technique in processing the data that is advantageous due to the unexpected contents tweets may have. An example of the block matching of tweets based on syntax is described below. To begin, the syntax of an incoming tweet is compared to following or stored previous tweets. When a set of tags of the same syntax is encountered the tweets are grouped together. If a tweet does not have the same set of tags, then a new group if formed. Therefore, eventually, all tweets up to some practical number are grouped.

Once a number of tweets have been collected the system will pass the blocks to the next step for additional classification. Otherwise, it will gather more tweets. To group based on grammatical structure, we developed a set of rules. For example, we looked for street names, and words like: blocked, open, and closed, to extract the meaning of a group. Then, the groups are further classified into groups that have a street name. Also, as the groups are constructed they will have more than one street segment depending on the cross street and intersection will could make a difference where some street segments occur. Thus, additional classification must be performed and sub-groups will be generated.

3 Example

3.1 Block matching

We chose a simple example to illustrate our approach. After eliminating non-traffic tweets and removing unwanted characters, the system starts by finding blocks of tags and grouping like patterns of blocks. We considered all plural and singular nouns as one type of tag. In our simple example, we used blocks that consisted of four tags because smaller blocks might miss important information, and larger groups add to the complexity. Correspondingly, to make sure these blocks have information about a place (street name or point of interest) we imposed a constraint that blocks should start with a word tagged as a NN, noun or NNP, proper noun. In addition, the tweets should contain a JJ, adjective or a VB, verb. We then followed the process illustrated in Fig. 1.

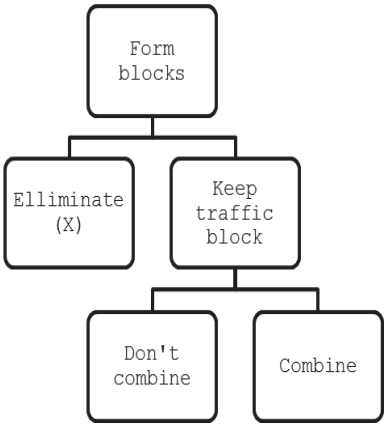


Figure 1 Creating blocks.

We imposed some rules on the forming of blocks. If a block did not contain a NNP tag, it was not considered. A block may be less than 4 tags if the tweet length is less than 4 tags. All blocks should start with (NNP) a proper noun Finally, the NNP will be examined with a gazetteer to determine either it's a street name or not.

Table 2 illustrates an example of how incoming tweets were extracted On the left were tweets after they have been tagged. Using our rules, we extracted the pertinent blocks and showed the results in the right column. In this example, we did not allow blocks to overlap, but each tweet could contain more than one block. Once blocks have been extracted, we had to still eliminate some of them based on additional rules..

No	POS tagging	No	Resulting blocks
1	NN/NNP/VBZ/JJ	1	NN/NNP/VBZ/JJ
2	NNP/IN/NNP	2	NNP/IN/NNP
3	NNP/NN/VBZ/JJ/NN/VBZ/JJ/ADV	3	NNP/NN/VBZ/JJ
4	V/NNP/IN/NNP/JJ/NN	4	NNP/ADJ/NN
5	NNP/NNP/NNP/VBZ/JJ/TO/NN	5	NNP/VBZ/JJ/TO/NN
6	ADJ/NNP/VBZ/JJ/ADV		
7	NNP/NNP/VBZ/JJ/TO/NN		
8	PRO/VBZ/DT/NN		
9	NN/VBZ/JJ		

Table 1 Extracting useful blocks from incoming blocks.

Three examples of the application of rules in our example is as follows. A tweet could consist of the sentence "Road is closed." However, the block of tags don't contain a street name. The block would be labeled as, " Road" NN/ "is" VBZ /"closed" JJ. This block has a noun but it is not a street name so it would be eliminated.

A block that does not contain a Verb tag, JJ will be elimated because that helps explain the status of the road or street. For example, a tweet could be "Accident on Street name." The blocks would be labeled as "Accident" NN / "ON" IN / "StreetName" NNP, and would be elliminated because the sentence does not have any information about whether the road or street is passable. We may be able to infer the condition of the road but did not do so in our simple example.

Finally, we showed a two block tweet of "Babcock closed" (where Babcock is a street name). Here, the block would be labeled as " Babcok" NNP / "Closed " JJ , and would be accepted

Table 3 below shows example of tweets and whether they rejected or accepted with the reason. Acceptable street names in the table are: Babcock, University, and 192.

Tweet	Action	reason
Emergency Babcock is closed	Accept	Gives information about status of street
Accident on Babcock	Reject	Doesn't give information about status of street
Babcock Road is Open	Accept	Gives info about status of street
Road is open	Reject	Doesn't have street name
Babcock around FIT major	Reject	Not useful information
Babcock closed expect delays	Accept	Gives information about status of street
FIT major accident	Reject	No information about street
Babcock/192 road closed	Accept	Gives information about status of street
Babcock and university open	Accept	Gives information about status of street
FIT major	Reject	Not useful information

Table 2 Example of accepted and rejected tweets.

The next step is concerned with creating an ideal number of blocks by either combining blocks or creating more of them. This step deals more with the language itself since it checks the type of tags and words. We used three combining rules in our example.

For tweets that have "NN" after the "NNP," the NN will be removed as it is most likely an extra descritor of a street name. For example the rule is written as NNP/NN/VBZ/JJ ➔ NNP/VBZ/JJ, and an example of its use is "Babcock street is closed" ➔ "Babcock is closed."

We also removed any VBZ tag that came between a noun and adjective because itusually doesn't add any information . For exapmle, the rule can be written as NNP/VBZ/JJ ➔ NNP/JJ, and an example of its use is "University is closed" ➔ "Univeristy closed."

For cases that have "RB" after the "VBZ" tag, we complemented the adjective or verb after the "RB" tab. The rule was written as NNP/VBZ/RB/JJ ➔ NNP/ JJ, and and example is, "Babcock is not closed" ➔ "Babcock is open."

In addtion to these rules, we used regular expressions to perform preprocessing on the tweets so they could be processed more efficiency. For example, we performed spell checking, removing of symbols, and other tasks.

In Fig. 2 a small portion of the dataset we've used and a sample of two of the created groups are shown. The original tweets are shown with their corresponding tags in the top half of the figure. In the bottom half, the useful tweet are listed in two groups that were formed. The groups were then further combined into one group by removing the NN tag. Then, two groups were created based on the street names. Finally, a decision was made for each street.

Figure 2 Creating groups

3.2 Decision

The final two steps consist of possibly splitting a group. For example in Fig. 3 there are two streets contained in the group, so the group would be split into two groups, one for each road.

The last step is concerned with making a decision. In this case we used a simple voting mechanism. For both streets, Babcock and Dairy, most of the tweets for each road indicated that the road is closed, therefore, those will be the outputs.

4 Conclusions

An effective method for determining the status of traffic on roads can be implemented by grouping blocks of Twitter tags in a data-driven approach. By grouping similar blocks together data can be extracted more efficiently that predetermined fixed rules. Then, groups can be further created or combined in useful patterns. Finally, a more accurate system may be able to be designed because a consensus from multiple tweets is used to determine the status of a road.

5 References

[1] http://www.Twitter.com

[2] Sakaki, T., Okazaki, M., and Matsuo, Y. 2010. Earthquake shakes Twitter users: real-time event detection by social sensors. In Proc. of the 19th international conference on World wide web (WWW 2010). pp. 851-860, 2010.

[3] Ribeiro, J., Sílvio, Davis, J., Clodoveu, Oliveira, D., Meira, J., Wagner, Gonçalves, T., & Pappa, G. (2012). Traffic observatory: A system to detect and locate traffic events and conditions using twitter. 5-11. doi:10.1145/2442796.2442800

[4] Wanichayapong, N., Pruthipunyaskul, W., Pattara-Atikom, W., & Chaovalit, P. (2011). Social-based traffic information extraction and classification. 107-112. doi:10.1109/ITST.2011.6060036

[5] Endarnoto, S. K., Pradipta, S., Nugroho, A. S., & Purnama, J. (2011). Traffic condition information extraction & visualization from social media twitter for android mobile application. 1-4. doi:10.1109/ICEEI.2011.6021743

[6] Marie-Catherine de Marneffe, Bill MacCartney and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In LREC 2006.

[7] http://www.link.cs.cmu.edu/link/index.html