IMMM 2015

The Fifth International Conference on Advances in Information Mining and Management

ISBN: 978-1-61208-415-2

DATASETS 2015

The International Symposium on Challenges for Designing and Using Datasets

June 21 - 26, 2015

Brussels, Belgium

IMMM 2015 Editors

Christian Ernst, Ecole des Mines de St Etienne - Gardanne, France
Andreas Schmidt, Karlsruhe University of Applied Sciences, Germany

IMMM 2015

Foreword

The Fifth International Conference on Advances in Information Mining and Management (IMMM 2015), held between June 21-26, 2015, in Brussels, Belgium, continued a series of academic and industrial events focusing on advances in all aspects related to information mining, management, and use.

The amount of information and its complexity makes it difficult for our society to take advantage of the distributed knowledge value. Knowledge, text, speech, picture, data, opinion, and other forms of information representation, as well as the large spectrum of different potential sources (sensors, bio, geographic, health, etc.) led to the development of special mining techniques, mechanisms support, applications and enabling tools. However, the variety of information semantics, the dynamic of information update and the rapid change in user needs are challenging aspects when gathering and analyzing information.

IMMM 2015 also featured the following Symposium:

- DATASETS 2015: The International Symposium on Challenges for Designing and Using Datasets

We take here the opportunity to warmly thank all the members of the IMMM 2015 Technical Program Committee, as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and efforts to contribute to IMMM 2014. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations, and sponsors. We are grateful to the members of the IMMM 2015 organizing committee for their help in handling the logistics and for their work to make this professional meeting a success.

We hope that IMMM 2015 was a successful international forum for the exchange of ideas and results between academia and industry and for the promotion of progress in the field of information mining and management.

We are convinced that the participants found the event useful and communications very open. We hope that Brussels, Belgium, provided a pleasant environment during the conference and everyone saved some time to enjoy the charm of the city.

IMMM 2015 Chairs:

IMMM Advisory Committee

Philip Davis, Bournemouth and Poole College - Bournemouth, UK
Andreas Schmidt, Karlsruhe University of Applied Sciences, Germany
David Newell, Bournemouth University - Bournemouth, UK
Kuan-Ching Li, Providence University, Taiwan
Abdulrahman Yarali, Murray State University, USA
Alain Casali, Aix Marseille Université, France

Ingrid Fischer, Universität Konstanz, Germany
Christian Ernst, Ecole des Mines de St Etienne - Gardanne, France
Paolo Garza, Politecnico di Torino, Italy
Bartlomiej Jefmanski, Wroclaw University of Economics, Poland
Nathalie Pernelle, Université Paris-Sud, France
Jürgen Pfeffer, Carnegie Mellon University, USA
Jörg Scheidt, University of Applied Sciences Hof, Germany
Ariella Richardson, Jerusalem College of Technology, Israel
Lorna Uden, Staffordshire University, UK
Eli Upfal, Brown University - Providence USA
Domenico Ursino, Università Mediterranea di Reggio Calabria, Italy
Jan Zizka, Mendel University - Brno, Czech Republic
Sung-Bae Cho (Chair), Yonsei University, Korea
Kyung-Joong Kim, Sejong University, Korea

IMMM Industry/Research Liaison Committee
Stefan Brüggemann, Astrium GmbH - Bremen, Germany
Olivier Caelen, Atos Worldline, Belgium
Feng Yan, Facebook Inc., USA
Katja Pfeifer, SAP AG, Germany
Arno H.P. Reuser, Reuser's Information Services, The Netherlands
Yulan He, Knowledge Media Institute / The Open University, UK
Mirela Danubianu, "Stefan cel Mare" University of Suceava, Romania
Wei Jin, Amazon.com, Seattle, USA
Olivier Caelen, Atos Worldline, Belgium
Yili Chen, Monsanto Company, USA
Gloria Bordogna, CNR - National Research Council / IDPA - Institute for the Dynamics of Environmental Processes - Dalmine, Italy
Daniel Kimming, Karlsruhe Institute of Technology, Germany
Josiane Mothe, IRIT, France
Dirk Labudde, Hochschule Mittweida, Germany
Yoseba Penya Landaburu, University of Deusto - Basque Country, Spain
Robert Wrembel, Poznan University of Technology, Poland

IMMM Publicity Chairs
Alessia Saggese, University of Salerno, Italy
Ludovico Boratto, Università di Cagliari, Italy
Toshio Kodama, University of Tokyo, Japan

IMMM 2015

COMMITTEE

IMMM Advisory Committee

Philip Davis, Bournemouth and Poole College - Bournemouth, UK
Andreas Schmidt, Karlsruhe University of Applied Sciences, Germany
David Newell, Bournemouth University - Bournemouth, UK
Kuan-Ching Li, Providence University, Taiwan
Abdulrahman Yarali, Murray State University, USA
Alain Casali, Aix Marseille Université, France
Ingrid Fischer, Universität Konstanz, Germany
Christian Ernst, Ecole des Mines de St Etienne - Gardanne, France
Paolo Garza, Politecnico di Torino, Italy
Bartlomiej Jefmanski, Wroclaw University of Economics, Poland
Nathalie Pernelle, Université Paris-Sud, France
Jürgen Pfeffer, Carnegie Mellon University, USA
Jörg Scheidt, University of Applied Sciences Hof, Germany
Ariella Richardson, Jerusalem College of Technology, Israel
Lorna Uden, Staffordshire University, UK
Eli Upfal, Brown University - Providence USA
Domenico Ursino, Università Mediterranea di Reggio Calabria, Italy
Jan Zizka, Mendel University - Brno, Czech Republic

IMMM Industry/Research Liaison Committee

Stefan Brüggemann, Astrium GmbH - Bremen, Germany
Olivier Caelen, Atos Worldline, Belgium
Feng Yan, Facebook Inc., USA
Katja Pfeifer, SAP AG, Germany
Arno H.P. Reuser, Reuser's Information Services, The Netherlands
Yulan He, Knowledge Media Institute / The Open University, UK
Mirela Danubianu, "Stefan cel Mare" University of Suceava, Romania
Wei Jin, Amazon.com, Seattle, USA
Olivier Caelen, Atos Worldline, Belgium
Yili Chen, Monsanto Company, USA
Gloria Bordogna, CNR - National Research Council / IDPA - Institute for the Dynamics of Environmental Processes - Dalmine, Italy
Daniel Kimming, Karlsruhe Institute of Technology, Germany
Josiane Mothe, IRIT, France
Dirk Labudde, Hochschule Mittweida, Germany
Yoseba Penya Landaburu, University of Deusto - Basque Country, Spain
Robert Wrembel, Poznan University of Technology, Poland

IMMM Publicity Chairs

Alessia Saggese, University of Salerno, Italy
Ludovico Boratto, Università di Cagliari, Italy
Toshio Kodama, University of Tokyo, Japan

IMMM 2015 Technical Program Committee

Aseel Addawood, Cornell University, USA
Zaher Al Aghbari, University of Sharjah, UAE
Riccardo Albertoni, Consiglio Nazionale delle Ricerche - Genova, Italy
César Andrés Sanchez, Universidad Complutense de Madrid, Spain
Annalisa Appice, Università degli Studi di Bari Aldo Moro, Italy
Avi Arampatzis, Democritus University of Thrace, Greece
Liliana Ibeth Barbosa Santillán, University of Guadalajara, Mexico
Shariq Bashir, National University of Computer and Emerging Sciences, Pakistan
Bernhard Bauer, University of Augsburg, Germany
Grigorios N. Beligiannis, University of Western Greece - Agrinio, Greece
Jorge Bernardino, ISEC - Polytechnic Institute of Coimbra, Portugal
Konstantinos Blekas, University of Ioannina, Greece
Jacek Blazewicz, Poznan University of Technology, Poland
Ludovico Boratto, Università di Cagliari, Italy
Gloria Bordogna, CNR - National Research Council / IDPA - Institute for the Dynamics of Environmental Processes - Dalmine, Italy
Stefan Brüggemann, Astrium GmbH - Bremen, Germany
Olivier Caelen, Atos Worldline, Belgium
Alain Casali, Aix Marseille Université, France
Mirko Cesarini, University of Milano Bicocca, Italy
Nadezda Chalupova, Mendel University - Brno, Czech Republic
Chi-Hua Chen, National Chiao Tung University, Taiwan R.O.C.
Weifeng Chen, California University of Pennsylvania, USA
Yili Chen, Monsanto Company, USA
Been-Chian Chien, University of Tainan, Taiwan
Sung-Bae Cho, Yonsei University, Korea
Kendra Cooper, University of Texas at Dallas, USA
Mirela Danubianu, "Stefan cel Mare" University of Suceava, Romania
Lois Delcambre, Portland State University, USA
Frantisek Darena, Mendel University - Brno, Czech Republic
Sébastien Déjean, Université de Toulouse & CNRS, France
Mustafa Mat Deris, University of Tun Hussein Onn, Malaysia
Emanuele Di Buccio, University of Padua, Italy
Qin Ding, East Carolina University - Greenville, USA
Mario Döller, University of Passau, Germany
Aijuan Dong, Hood College - Frederick, USA
Nikolaos Doulamis, National Technical University of Athens, Greece
Anass Elhaddadi, University of Paul Sabatier - Toulouse, France
Christian Ernst, Ecole des Mines de St Etienne - Gardanne, France
Manuel Filipe Santos, University of Minho, Portugal
Ingrid Fischer, Universität Konstanz, Germany
Rita Francese, Università degli studi di Salerno, Italy

Mehmet Tan, TOBB University of Economics and Technology, Turkey
Yi Tang, Chinese Academy of Sciences, China
Xiaohui (Daniel) Tao, The University of Southern Queensland, Australia
Olivier Teste, Université de Toulouse, France
Daniel Thalmann, Institute for Media Innovation (IMI) - Nanyang Technological University, Singapore
Alberto Tonda, UMR 782 GMPA - INRA, France
Michael Tschuggnall, University of Innsbruck, Austria
Vincent S. Tseng, National Cheng Kung University, Taiwan, R.O.C.
Chrisa Tsinaraki, European Union - Joint Research Center (JRC), Italy
Pavel Turcinek, Mendel University - Brno, Czech Republic
Franco Turini, University of Pisa, Italy
Lorna Uden, Staffordshire University, UK
Eli Upfal, Brown University - Providence USA
Domenico Ursino, Università Mediterranea di Reggio Calabria, Italy
Julien Velcin, Université de Lyon 2, France
Corrado Aaron Visaggio, University of Sannio, Italy
Zeev Volkovich, ORT Braude College Karmiel, Israel
Stefanos Vrochidis, Information Technologies Institute - Centre for Research and Technology Hellas, Greece
Baoying (Elizabeth) Wang, Waynesburg University, USA
Qi Wang, University of Science and Technology of China, China
Alexander Wöhrer, Vienna Science and Technology Fund, Austria
Hao Wu, Yunnan University - Kunming, P.R.China
Feng Yan, Facebook Inc., USA
Chao-Tung Yang, Tunghai University, Taiwan
Zhenglu Yang, University of Tokyo, Japan
Kui Yu, School of Computing Science - Simon Fraser University, Canada
Jan Zizka, Mendel University - Brno, Czech Republic

DATASETS 2015 Advisory Committee

Sung-Bae Cho (Chair), Yonsei University, Korea
Kyung-Joong Kim, Sejong University, Korea

DATASETS 2015 Program Committee Members

Francisco Henrique Cerdeira Ferreira, Universidade Federal de Juiz de Fora, Brazil
Yun-Maw Kevin Cheng, Tatung University, Taiwan
Sung-Bae Cho (Chair), Yonsei University, Korea
Yun Jang, Sejong University, Korea
Katarzyna Kaczmarek, Polish Academy of Sciences-Warsaw, Poland
Rene Kaiser, JOANNEUM RESEARCH, Austria
Kyung-Joong Kim, Sejong University, Korea
Irwin King, Chinese University of Hong Kong, China
Thomas Larsson, Mälardalen University-Västerås, Sweden
Henning Müller, HES-SO Valais, Switzerland
Unil Yun, Sejong University, Korea
Matthias Zeppelzauer, St. Pölten University of Applied Sciences, Austria

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

Closed Frequent Itemset Mining over Fast Data Stream Based on Hadoop

Shan Jicheng, Liu Qingbao
Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
email: sjcheng2007@126.com, liuqingbao@nudt.edu.cn

***Abstract*—Mining closed frequent itemsets provides complete and condensed information for non-redundant association rules generation. Online mining of closed frequent itemsets over streaming data is one of the most important issues in mining data streams. In this paper, we extend two types of methods to MapReduce platform to mine closed frequent itemset over fast data streams. Experiments show that both methods have performance improvement with more mapper nodes and the vertical format data method has higher speed to process fast data streams.**

Keywords- data stream; closed frequent itemsets; mapreduce.

I. INTRODUCTION

Frequent itemset mining has been an important research issue for many years in data mining community. With the development of data storage and data processing, frequent itemset mining meets new challenges and needs to be extended. For example, Wireless Sensor Network (WSN) can be used to monitor the traffic status and the environment information. With time flows, the WSN will produce a large scale of data that cannot be storaged in traditional static database. WSN data related to time should be processed as stream data with special methods. However, most of the data stream mining methods face the performance problem as they are often used on one computer which has poor computing ability. When the stream becomes 'bigger' and 'faster', these methods have lower effect or even cannot work.

For mining frequent itemsets in traditional transactional database, Apriori is the most classic and most widely used algorithm proposed by R. Agrawal and R. Srikant in 1994 [1]. The algorithm works in a multi-phase generation-and-test framework, including the joining and pruning process to reduce the number of candidates before scanning the database for frequency computing. The algorithm terminates when no more candidate itemsets can be generated. The Apriori levelwise approach implies several scans over the database for support counting of candidate itemsets which affects the performance of the algorithm. To reduce the scan overhead, some depth-first methods were proposed, of which the Eclat (Equivalent CLASS Transformation) algorithm by Zaki [2] and the FP-Growth algorithm by Han, Pei, Yin, and Mao [3] are typical representatives. These algorithms use compressed data structure to store necessary transaction information and avoid candidate generation and levelwise scans.

Recently, the increasing emergence of data streams has led to the study of online mining of frequent itemsets, which is an important technique for a wide range of emerging applications [4], such as web search and click-stream mining, trend analysis and fraud detection in telecommunications data, e-business and stock market analysis, and wireless sensor networks. Unlike mining static databases, mining data streams poses many new challenges. Firstly, it is not realistic to store the whole data stream in the main memory or even the secondary storage space as the data continuously come with no boundary. Secondly, traditional methods working on static stored datasets by multiple scans are unrealistic, since the streaming data is passed only once. Thirdly, stream mining requires highly efficient real-time processing in order to keep up with the high data arrival rate and mining results are expected to be available within short response time. In addition, the combinatorial explosion of itemsets exacerbates mining frequent itemsets over streams in terms of both memory consumption and time expense. In the past ten years, many algorithms to mine frequent itemsets over data stream have been proposed, like Lossy Counting [5], DSM-FI [6], FDPM [7] , estDec [8], FP-streaming [9], estWin [10], Moment [11], etc. These algorithms can be divided into two categories based on the window they adopt: the landmark window model and the sliding window model.

With the advent of Internet and the exponential growth of data volume towards a terabyte or more, it has been more difficult to mine them on a single sequential machine. Researchers attempt to parallelize these frequent itemset mining algorithms to speed up the mining of the ever-increasing sized databases. In big data era, we need new framework and new methods to capture and deal with dynamic changing, high dimensional, large scale data. In 2004, Google proposed their Google File System [12] and MapReduce [13] framework which has been successfully used in Google search and other Google products. With some number of ordinary computers, Google Distributed File System solved the big data storage problem and MapReduce framework can be used to do computing work on the big data stored. In a MapReduce cluster, a node which schedules tasks execution among nodes is called the master, and other nodes are workers. MapReduce uses two phase procedure to implement Function Programming, map and reduce. The master is responsible for the scheduling of the map tasks and

 ISBN: 978-1-61208-415-2

the reduce tasks which are executed by the workers after the job is initialized. In Map phase, the map function in each node takes the input data as <key, value> pair and outputs a list of <key, value> pairs in different domain. Then in Reduce phase, the reduce function in nodes takes the output of map functions as <key, list-of-values> and outputs a collection of values as the result. Also, the output of the reduce function can be formatted as <key, value> pairs which makes multiphase mapreduce iteration possible. What's more important, both the map and reduce functions can be performed in parallel.

MapReduce hides the problems like fault tolerance, data distribution and load balancing in parallelization, which allows user to focus on the computing implementation problem without worrying about the parallelization details. Developers only need to write the map function to read blocks from the distributed file system and produce a set of intermediate <key, value> pairs. The MapReduce framework organizes together all intermediate values related to the same intermediate key , often with a shuffle procedure, and sends them to the reduce function [13]. The reduce function, also written by the user, captures an intermediate key and a set of values for that key. Then reduce function merges together these values to produce an aggregate result. This merging allows users to handle lists of values that are too large to fit in memory. Thus, MapReduce can be an efficient platform for mining frequent itemsets from huge datasets of tera- or peta-bytes [14][15][16][17][18].

In this paper, we consider to mining closed frequent itemsets over data stream with sliding window model based on the MapReduce framework. Closed frequent itemsets can store necessary information to get complete frequent itemsets with less storage requirement [19]. Sliding window model pays different attention to data produced at different time so that it can discover time-related rules which are more important in stream application environment. Based on the MapReduce framework, our method has higher performance and ability to process high-velocity large-volume dynamic-variety stream data.

The rest of this paper is organized as follows. The preliminary knowledge is given in Section II. Section III describes details of the two methods we extend and implement on MapReduce platform. Experiment results are shown and analyzed in Section IV. We conclude in Section V.

II. PRELIMINARIES

Let $A = \{a_1, \ldots, a_m\}$ be a set of **items**. Items may be commodities, products, records, internet links etc. Any subset $I \subseteq A$ is called an itemset. Let $T = (t_1, \ldots, t_n)$ be a set of **transactions** within a slide window of size *n* denoted by data stream. Each unique transaction t_i of *T* is a pair <tid_i, $k\text{-}items_i$> of which $k\text{-}items_i \subseteq A$ is a set of *k* items. A transaction database can list, for example, the sets of products bought by the customers of a supermarket within a period of time, or the sets of pages a user visited for a site in a session. Every transaction refers to an itemset, but some itemsets may not appear in *T*.

Let $I \subseteq A$ be an itemset and *T* a transaction database over *A*. A transaction $t \subseteq T$ **covers** the itemset *I* or the itemset is **contained in** transaction t if and only if $I \subseteq t$.

The set $K_T(I) = \{k \in \{1, \ldots, n\} | I \subseteq t_k\}$ is called the **cover** of *I* w.r.t. *T*. The cover of an itemset is the index set of transactions that cover it.

The value $s_T(I) = |K_T(I)|$ is called the **absolute support** of *I* with respect to *T*. The value of $\sigma_T(I) = \frac{1}{n}|K_T(I)|$ is called the **relative support** of *I* with respect to *T*. The support of *I* is the number or fraction of transactions that cover it. Sometimes $\sigma_T(I)$ is also called the (**relative**) **frequency** of *I* in *T*.

The Frequent Itemset Mining problem can be formally defined as:

- Given:
  - a set $A = \{a_1, \ldots, a_m\}$ of items;
  - a vector $T = (t_1, \ldots, t_n)$ of transactions over *A*;
  - a number σ_{min} such that $0 < \sigma_{min} < 1$, the **minimum support**.
- Goal:
  - the set of frequent itemsets, that is, the set $\{I \subseteq A \mid \sigma_T \geq \sigma_{min}\}$.

As shown in Figure 1, all the frequent k-itemsets (k=1,2,3) for the transaction database *T* left with 10 transactions are listed right given the minimum support s_{min}= 3. So the frequent itemset for *T* is

$\mathcal{F} = \{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{a,c\}, \{a,d\}, \{a,e\}, \{b,c\}, \{c,d\}, \{c,e\}, \{d,e\}, \{a,c,d\}, \{a,c,e\}, \{a,d,e\}\}$

According to the priori property, every subset of a frequent itemset is also frequent. Thus, generation-and-test algorithms to mine all frequent itemsets (complete frequent itemsets) suffer from the problem of combinatorial explosion. To solve this problem two substitute solutions have been proposed. In the first solution, only maximal frequent itemsets are mined. A frequent itemset is maximal if none of its superset is frequent. The number of maximal frequent itemsets $\mathcal{M}$ is usually smaller than the number of complete frequent itemsets $\mathcal{F}$, and we can derive all the members of $\mathcal{F}$ from $\mathcal{M}$. It is a pity that $\mathcal{M}$ does not contain

TID	Itemset
1	{b,c,d}
2	{a,d,e}
3	{a,c,d,e}
4	{a,c,e}
5	{a,c,d}
6	{a,e}
7	{a,c,d,e}
8	{b,c}
9	{a,d,e}
10	{b,c,e}

1 item	2 items	3 items
{a}:7	{a,c}:4	{a,c,d}:3
{b}:3	{a,d}:5	{a,c,e}:3
{c}:7	{a,e}:6	{a,d,e}:4
{d}:6	{b,c}:3	
{e}:7	{c,d}:4	
	{c,e}:4	
	{d,e}:4	

Figure 1. A transaction database, with 10 transactions, and the enumeration of all possible frequent itemsets using the minimum support of s_{min}= 3 or σ_{min} = 0.3 = 30%.

 ISBN: 978-1-61208-415-2

support information of itemsets that do not belong to $\mathcal{M}$. Thus, discovering only maximal frequent itemset loses information.

The second solution maintains enough information to get complete frequent itemsets. It discovers all closed frequent itemsets from the database. An itemset is closed if and only if none of its superset has the same support as it has. Similarly, the number of closed frequent itemsets $\mathcal{C}$ is smaller than that of $\mathcal{F}$. More importantly, we can derive $\mathcal{F}$ from $\mathcal{C}$ because a frequent itemset I must be a subset of one or more closed frequent itemset, and I's support is equal to the maximal support of the closed itemsets it is contained in.

For the three kinds of frequent itemsets, $\mathcal{F}, \mathcal{M}, \text{and } \mathcal{C}$, we can get their relation which is $\mathcal{M} \subseteq \mathcal{C} \subseteq \mathcal{F}$. The maximal and closed frequent itemsets for the example above are:

$$\mathcal{C} = \{(a,7),(c,7),(e,7),(d,6),(ae,6),(ad,5),(ade,4),(ac,4),(cd,4),(ce,4),(acd,3),(ace,3),(bc,3)\}$$

$$\mathcal{M} = \{(acd,3),(ace,3),(ade,4)\}$$

Since $\mathcal{C}$ is smaller than $\mathcal{F}$ with no information loss about any frequent itemset, in this paper, we focus on the closed frequent itemsets mining.

III. DATA STREAM MIMING ON MAPREDUCE

We designed two methods to mine high speed data streams on Hadoop platform and to make a comparison. In both solutions, we compress the high velocity data and split it into basic blocks. Every single block is a basic window unit processed by a mapper node. For the first method, we modified the moment algorithm to fit the MapReduce framework: as data flows in, single transactions are added to FP-Tree structure to maintain the data information. When the number of transactions reaches the threshold, the Closed Enumeration Tree (CET), which will be explained in Section IIIA, will be built for the first time. Then, the new transaction continues to be added and old transaction is deleted causing update of the CET. CET maintains enough information to get the closed frequent itemsets for the data stream at any moment. For the second method, we use vertical format data to store the item and transaction information. We build a matrix for basic window units. Every item contained in the stream has a line vector which lists all the transaction identifiers cover this item. Then, we can build itemset following alphabet order with item's transaction cover vector. As computer has superiority of vector computing, the support counting and closure judgment will be easier. In Section V, we show the implementation and experiment results of the two methods on synthetic and real datasets.

A. *Moment-based MapReduce mining*

Moment[11] was used to update closed frequent itemsets for sliding window incrementally. It adopted a prefix tree structure in main memory called Closed Enumeration Tree (CET) to maintain the itemsets selected from the sliding window dynamically. The CET contains four node types which were described in detail in [11]. They are Infrequent Gateway Nodes (IGN), Unpromising Gateway Nodes (UGN), Intermediate Nodes (IN) and Closed Nodes (CN). Figure 2 shows an example of a sliding window and its CET structure in which dashed circle represents IGN, dashed rectangle represents UGN, solid circle represents IN and CN is represented by solid rectangle. From the Apriori property, all super sets of infrequent itemset are not frequent, we can get: IGN has no super set in the CET, child nodes of UGN cannot be CN so that we do not need to maintain child nodes of UGN. CET only needs to store small part of the itemsets still being able to get accurate results.

When a new transaction arrives, Moment explores nodes related to the transaction in the CET. For every node explored, Moment increase the support count and update the node type. In Figure 3, a new transaction T (tid 5) is added to the sliding window. We traverse the parts of the CET that are related to transaction T. For each related node nI , we update its support, tid sum, and possibly its node type.

When an old transaction is to be deleted, Moment also explores nodes related to the transaction in the CET. For every node explored, Moment decreases the support count and updates the node type. In Figure 4, an old transaction T (tid 1) is deleted from the sliding window. To delete a transaction, we also traverse the parts of the CET that is related to the deleted transaction. For each related node nI , we update its support, tid sum, and possibly its node type.

For its incremental way of updating for window's sliding, Moment has a formally process procedure and becomes fundamental method to mine closed frequent itemsets for data stream.

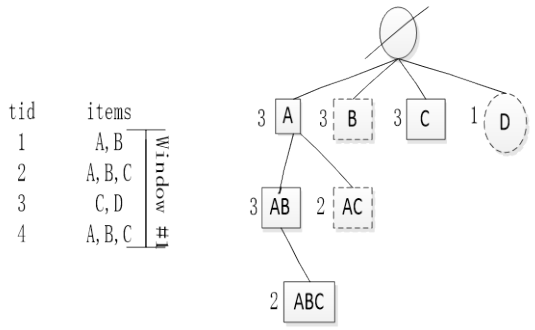


Figure 2. The Closed Enumeration Tree Corresponding to Window #1

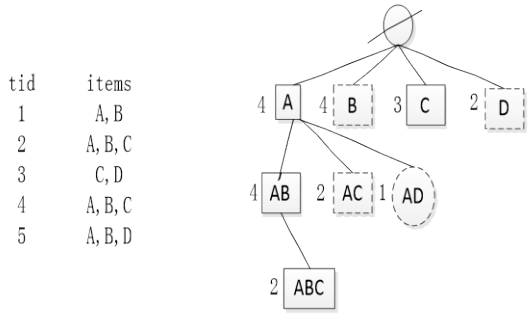


Figure 3. Adding a transaction

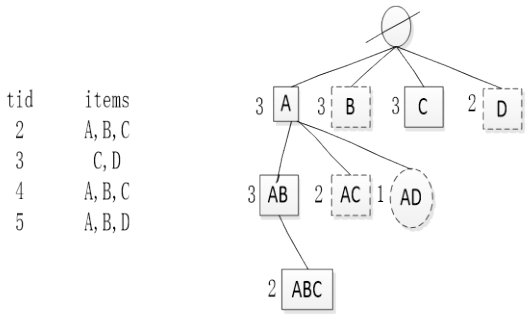


Figure 4. Deleting a transaction

The MapReduce implementation of Moment M-Moment updates the whole CET using basic window as unit. The stream receiving modular compress the data stream to transaction format that can be produced as the input of the map function. A basic window was the split for a mapper node. The mapper node mines the split unit using Moment algorithm and send the intermediate results to reducer node. The reducer node calls the reduce function to combine intermediate key-value pairs to get the whole result as the closed frequent itemsets currently for the data stream.

B. Vertical format data based MapReduce mining

Vertical format was often used in Eclat-like methods. The transaction database was transformed to item-transaction matrix. The matrix was built with tid-list rows. A tid-list consists of two fields: *Item* and *Tidset* field. The *Tidset* for an item i_p is denoted as *tidset*(i_p) which is a set of transaction identifiers containing item i_p. *Tidset* is a set structure that makes the *find(tid)* and *inter(tidset1,tidset2)* easy to implement and execute. Furthermore, we use an extended prefix tree to list itemsets with support and a hash table storing all closed frequent itemsets with their support as keys to check a new frequent itemset is closed or not.

In the following, we discuss the related algorithms to deal with window moving [20]. All the algorithms have the same input parameters (*nI*, *N*, *s*) and result in the updating of the itemset type and the hash table. *nI* stands for the item to deal with, *N* is the window size, and *s* means the relative support threshold. Figure 5 describes the algorithm to build the hash table. In the building algorithm, each n_I has a corresponding *tidset*, *Tidset(I)*, to store the transactions information in the current sliding window. Function Build is a depth-first procedure. Build visits the itemsets in a lexicographical order. In the lines 1–2 of the algorithm, function Build is performed if n_I is frequent and is not contained by other closed frequent itemsets. Function *leftcheck* uses the support of n_I as a hash key to speed up the checking. In the lines 3–5, if n_I passes the checking of the lines 1–2, Build generates all possible children of n_I with frequent siblings and creates their tidset by set intersect operation of n_I its frequent siblings. In the lines 6–7, Build recursively calls itself to check each child of n_I .In the lines 8–10, if there is no child of n_I with the same support as n_I, n_I is a closed frequent itemset and it is retained in the hash table.

```
Build (n_I, N, s)
1:   if support(n_I) = s*N then
2:       if leftcheck(n_I) = false then
3:           foreach frequent sibling n_K of n_I do
4:               generate a new child n_{I∪K} for n_I ;
5:               intersect Tidset(I) and Tidset(K) to obtain Tidset(I ∪K)
6:           foreach child n_I' of n_I do
7:               Build(n_I', N, s);
8:           if no child n_I' of n_I such that support(n_I') = support(n_I) then
9:               retain n_I as a closed frequent itemset;
10:              insert n_I into the hash table;
```

Figure 5. Algorithm of Build

When continues to read transactions after the window is full, the window slides with two operations: delete the oldest transaction and append the new incoming transaction.

Deleting the oldest transaction is the first step of window sliding. First of all, all items are visited to check if the deleted transaction contains it. Then, all items in the deleted transaction are kept and corresponding transaction id is deleted from their tidsets. Figure 6 gives the algorithm of deleting the oldest transaction after removing the transaction id from the tidsets of items. In Figure 6, the function *Delete* generates the prefix tree including the itemsets whose supports are beyond s*N - 1. This is because the supports of a set of closed frequent itemsets in previous window would be s*N and then becomes s*N - 1 after the deletion. If n_I is a closed frequent itemset, the hash table is updated. In the lines 19 and 23, if n_I is closed frequent itemset in previous window, n_I is marked as a non-closed itemset. In this case, n_I will not be retained when the function Delete is done.

Appending the incoming transaction is the second step of window sliding. The new transaction id will be added to the tidset of the items which are contained in the transaction.

```
Delete(n_I, N, s)
1:   if n_I is not relevant to the deleted transaction then
2:       return;
3:   else if support(n_I) = (s*N − 1) then
4:       foreach sliding n_K of n_I whose support = (s*N − 1) do
5:           generate a new child n_{I−K} for n_I ;
6:           intersect Tidset(I) and Tidset(K) to obtain Tidset(I ∪K)
7:       foreach child n_I' of n_I do
8:           Delete(n_I', N, s);
9:       if support(n_I) = (s*N) then
10:          if leftcheck(n_I) = false then
11:              if n_I is closed frequent itemset in previous window then
12:                  update the support of n_I;
13:                  update n_I in the hash table;
14:              else
15:                  retain n_I as a closed frequent itemset;
16:                  insert n_I into the hash table;
17:          else
18:              if n_I is closed frequent itemset in previous window then
19:                  mark n_I as a non-closed frequent itemset;
20:                  remove n_I from the hash table;
21:      else
22:          if n_I is closed frequent itemset in previous window then
23:                  mark n_I as a non-closed itemset;
24:                  remove n_I from the hash table;
```

Figure 6. Algorithm of Delete

```
Append (n_I, N, s)
1:    if support(n_I) = s*N then
2:        if leftcheck(n_I) = false then
3:            foreach frequent sibling n_K of n_I do
4:                generate a new child n_{I∪K} for n_I ;
5:                intersect Tidset(I) and Tidset(K) to obtain Tidset(I ∪K)
6:            foreach child n_I' of n_I do
7:                Append (n_I', N, s);
8:            if no child n_I' of n_I such that support(n_I') = support(n_I) then
9:                if n_I is closed frequent itemset in previous window then
10:                   update the support of n_I;
11:                   update n_I in the hash table;
12:               else
13:                   retain n_I as a closed frequent itemset;
14:                   insert n_I into the hash table;
```

Figure 7. Algorithm of Append

Figure 7 gives the algorithm of appending a new incoming transaction after the tidset adding. Function Append is almost the same as Build. The only difference is in the lines 9–11. If the checked closed frequent itemsets are already in the hash table, Append updates the hash table.

The MapReduce implementation M-vertical is similar to the content described in Section III.A.

IV. EXPERIMENTAL RESULT

In this section, we evaluate the performance of the MapReduce implementation of the two methods and make comparison between them. The Java source code of the essential version of Moment is downloaded from the open source site www.admire-project.eu (by Maciek Jarka), and Java source code of a method use vertical format data to mine frequent itemsets is derived from [21]. The Hadoop version is 1.2.1. All experiments are done on a cluster of computers with 2GB memory and Pentium（R）Dual CPU E2200@2.20GHz running on Ubuntu 12.04 OS. We generate a synthetic dataset T10I4D100K from IBM data generator [1]. The parameters are described as follows: T is average transaction size; I is average size of maximal potential frequent itemsets; D is the total number of transactions. Besides, a real-world dataset Mushroom was downloaded from FIMI Repository [22].

A. *Mining with different minimum supports*

In the first experiment, the minimum support threshold is changed from 1% to 0.1%, and the size of the sliding window is fixed to 1000 transactions.

Figure 8 shows the loading time of the first window. In the first window, both methods need to build a prefix tree. It can be observed that the vertical based method is faster than M-Moment. It is because that generating candidates and counting their supports with vector set is more efficient.

Figure 9 shows the average time to process single transaction when window slides. It also shows that the later method is faster for similar reason. When Moment slide the window, the adding and deleting of transaction cause explore of the tree structure. However, when M-vertical method slides, the algorithm only visit items that the added or deleted transaction contains and the updating of the hash table is very fast.

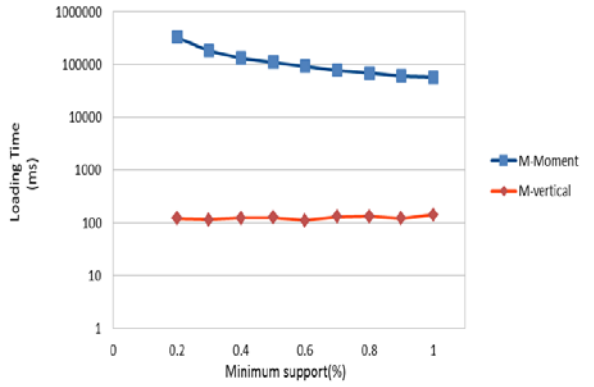


Figure 8. Time of loading the first window with different minimum supports

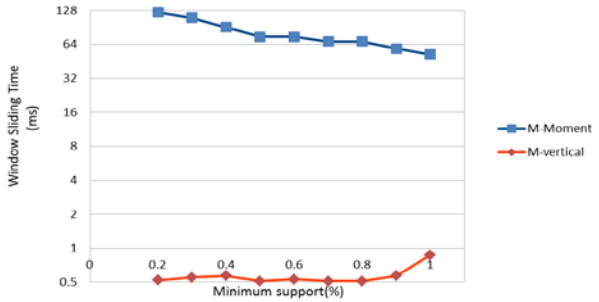


Figure 9. Average time of window sliding with different minimum supports

Because of the vertical format data structure, it can also be seen that the metric change extent of the latter method is not as much as the former one.

B. *Mining with different number of mappers*

In this experiment, the number of mappers for the two methods is changed from 1 to 10. The size of the basic window is fixed to 10000 and minimum support threshold is set to 0.1%. Figure 10 shows the total execution time to process 100000 transactions with the two methods.

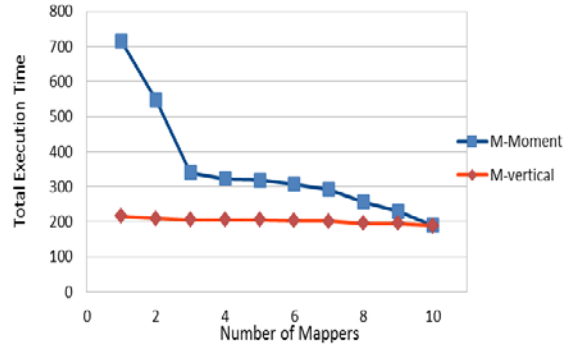


Figure 10. Total execution time with different number of mappers

It can be seen that for the MapReduce Moment method when mapper nodes increase the total time decrease a lot. For the Mapreduce vertical method, the total execution time also decreases a little as the number of mappers increase. But the change is not as obvious as the M-Moment. We can conclude that with more mapper nodes, the ability of both methods improves.

V. CONCLUSION AND FUTURE WORK

In this paper, we extend and implement two types of methods to do experiments on Hadoop platform to mine closed frequent itemset over fast data streams. We firstly use CET structure and Moment algorithms to mining. Then, we introduce vertical format data to maintain item-transaction information. Experiments show that vertical format data method has higher speed and performance to process fast data streams. Through extend implementation on Hadoop we observed that increasing number of mappers can improve both methods' ability to face up with fast data streams. As for the future work, we consider to design new methods fitting MapReduce better and to do experiments on cluster with more nodes to make the results more clear.

ACKNOWLEDGMENT

We thank Maciek Jarka and Sandy Moens and team for sharing their codes online.

REFERENCES

[1] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," Proc. 20th int. conf. very large data bases, VLDB. vol. 1215, Sep. 1994, pp. 487-499.

[2] Zaki and M. Javeed, "Scalable algorithms for association mining," Knowledge and Data Engineering, IEEE Transactions, Dec. 2000, pp. 372-39, doi:10.1109/69.846291.

[3] J. Han, J. Pei, and Y. Yin. "Mining frequent patterns without candidate generation." ACM SIGMOD Record. vol. 29, May. 2000, pp. 1-12, doi:10.1145/342009.335372.

[4] M. Garofalakis, J. Gehrke, and R. Rastogi, "Querying and mining data streams: you only get one look a tutorial,". In SIGMOD Conference , vol. 2002, Jun. 2002, p. 635, doi:10.1145/564691.564794.

[5] G. S. Manku and R. Motwani, "Approximate frequency counts over data streams," In Proceedings of the 28th international conference on Very Large Data Bases, Aug. 2002, pp. 346-357, doi:10.14778/2367502.2367508.

[6] H. F. Li, S. Y. Lee, and M. K. Shan, "An efficient algorithm for mining frequent itemsets over the entire history of data streams," In Proc. of First International Workshop on Knowledge Discovery in Data Streams, Sep. 2004.

[7] J. X. Yu, Z. Chong, H. Lu, and A. Zhou, "False positive or false negative: mining frequent itemsets from high speed transactional data streams," In Proceedings of the Thirtieth international conference on Very large data bases, vol. 30, Aug. 2004, pp. 204-215.

[8] J. H. Chang and W. S. Lee, "Finding recent frequent itemsets adaptively over online data streams," In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, Aug. 2003, pp. 487-492, doi:10.1145/956750.956807.

[9] C. Giannella, J. Han, J. Pei, X. Yan, and P. S. Yu, "Mining frequent patterns in data streams at multiple time granularities," Next generation data mining, 2003, pp. 191-212.

[10] J. H. Chang, and W. S. Lee, "estWin: adaptively monitoring the recent change of frequent itemsets over online data streams," In Proceedings of the twelfth international conference on Information and knowledge management, Nov. 2003, pp. 536-539, doi:10.1145/956863.956967.

[11] Y. Chi, H. Wang, P. S. Yu, and R. R. Muntz, "Moment: Maintaining closed frequent itemsets over a stream sliding window," In Data Mining, 2004, ICDM'04, Fourth IEEE International Conference on, Nov. 2004, pp. 59-66, doi:10.1109/ICDM.2004.10084.

[12] S. Ghemawat, H. Gobioff, and S. T. Leung, "The Google file system," In ACM SIGOPS operating systems review , vol. 37, No. 5, Oct 2003, pp. 29-43, doi:10.1145/1165389.945450.

[13] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," Communications of the ACM, Vol 51, Jan. 2008, pp. 107-113, doi:10.1145/1327452.1327492.

[14] Z. Farzanyar, and N. Cercone, "Accelerating Frequent Itemsets Mining on the Cloud: A MapReduce-Based Approach," In Data Mining Workshops (ICDMW), IEEE 13th International Conference on, Dec. 2013, pp. 592-598, doi:10.1109/ICDMW.2013.106.

[15] Z. Farzanyar, and N. Cercone, "Efficient mining of frequent itemsets in social network data based on MapReduce framework," In Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Aug. 2013, pp. 1183-1188, doi:10.1145/2492517.2500301.

[16] F. Kovacs, and J. Illés, "Frequent itemset mining on hadoop," In Computational Cybernetics (ICCC), 2013 IEEE 9th International Conference on, July. 2013, pp. 241-245, doi:10.1109/ICCCyb.2013.6617596.

[17] H. Chen, T. Y. Lin, Z. Zhang, and J. Zhong, "Parallel mining frequent patterns over big transactional data in extended mapreduce," In GrC , Dec. 2013, pp. 43-48, doi:10.1109/GrC.2013.6740378.

[18] X. Wei, Y. Ma, F. Zhang, M. Liu, and W. Shen, "Incremental FP-Growth mining strategy for dynamic threshold value and database based on MapReduce," In Computer Supported Cooperative Work in Design (CSCWD), Proceedings of the 2014 IEEE 18th International Conference on, May 2014, pp. 271-276, doi:10.1109/CSCWD.2014.6846854.

[19] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal, "Discovering frequent closed itemsets for association rules," In Database Theory—ICDT'99 , vol. 1540, Jan. 1999, pp. 398-416, doi:10.1007/3-540-49257-7_25.

[20] H. F. Li, C. C. Ho, and S. Y. Lee, "Incremental updates of closed frequent itemsets over continuous data streams," Expert Systems with Applications, vol. 36, Mar. 2009, pp. 2451-2458, doi:10.1016/j.eswa.2007.12.054.

[21] S. Moens, E. Aksehirli, and B. Goethals, "Frequent itemset mining for big data," In Big Data, 2013 IEEE International Conference on, Oct. 2013, pp. 111-118, doi:10.1109/BigData.2013.6691742.

[22] Frequent Itemset Implantation Repository(FIMI), http://fimi.cs.helsinki.fi/,[retrieved: May, 2015].

 ISBN: 978-1-61208-415-2

Robustness of Bisecting *k*-means Clustering-based Collaborative Filtering Algorithm

Alper Bilge and Huseyin Polat
Computer Engineering Department
Anadolu University
Eskisehir, Turkey
emails: {abilge, polath}@anadolu.edu.tr

***Abstract*—The unprecedented popularity of e-shopping amenities provided by online retailers escalates attention to recommendation facilities. Collaborative filtering is one of the well-known recommendation techniques that helps customers choose possible products of interest by automating word-of-mouth habits. However, due to their nature, recommendation algorithms are open to shilling attacks of malicious users to promote/demote certain products. We propose bisecting *k*-means clustering-based recommendation algorithm as a robust algorithm in non-private environments against well-known shilling attacks. We investigate its robustness against shilling attacks by performing real databased experiments. We also analyze the effects of varying attacking parameters. We empirically establish that the algorithm is resilient against shilling attacks without significantly influenced by malicious profiles.**

Keywords–robustness; shilling; clustering; recommendation.

I. INTRODUCTION

With increasing amount of information available in everyday life through widespread use of the Internet, Collaborative Filtering (CF) systems have become one of the most practical tools to determine useful information. Such systems are very successful to cope with information overload problem. CF algorithms are efficient in automating word-of-mouth habits of individuals by collecting preference information about products such as movies, music CDs, books, and so on. Typically, CF systems hold a user-item matrix containing ratings of users on products and whenever a user requests for a prediction on a target product, the system produces an estimation as a weighted average of similar users' ratings on the target product.

CF systems are usually unable to strictly distinguish genuine profiles from malicious ones. Thus, they are vulnerable to potential manipulations. Either malicious users or competing companies might intrude bogus profiles into the database in order to favor or disfavor a certain product's popularity [1]. Such intrusions are called shilling attacks, which can be categorized as push or nuke attacks according to their intent [2]. Determining fake profiles and being robust against them is critical for the success of CF algorithms. Shilling attacks have been shown to be very effective against traditional memory based CF schemes [3][4]. However, clustering-based approaches are successful in distinguishing shilling profiles from genuine ones because bogus profiles expose high resemblance among themselves, which makes them to be clustered mostly together [1][5][6]. Hence, clustering-based methods are preferable over other schemes in order to achieve required level of robustness in CF systems.

There are some common requisites of CF systems such as accuracy, scalability, and robustness. A qualified CF algorithm is required to produce personalized predictions with decent accuracy to please customers and increase online sales. Moreover, due to constantly enlarging dimensions of user-item matrix, such algorithms should be resistant against scalability issues. Finally, it is expected for the algorithms not to be significantly affected by shilling attacks and be robust against them arising from their data collection nature. In the literature, there are various techniques developed to enhance quality of produced predictions by modifying similarity calculation methods [7] and handling sparse user profiles [8]. Some researchers proposed several CF algorithms to overcome scalability issues using matrix factorization [9][10], dimensionality reduction [11][12], and clustering techniques [13][14]. And finally, some model-based techniques have been shown to be resistant against shilling attacks [1][4].

Although the essential constraints of CF systems are discovered, it is hard to claim that there exists an eligible CF algorithm fulfilling all of them. Memory-based CF schemes are very successful in producing high quality referrals. However, they suffer from scalability issues and they are vulnerable to shilling attacks [15]. Model-based and hybrid CF methods are generally scalable and more resistant against shilling attacks; however, they commonly compromise from accuracy and often come with high computational cost for model update [1][16].

A scalable, low cost, and easy-to-interpret CF algorithm is proposed to produce highly accurate predictions in both non-private and privacy-preserving CF environments [17]. Its robustness against shilling attacks in private environments is also investigated [18]. However, such algorithm is not investigated with respect to robustness in non-private environments. Since clustering-based CF algorithms are successful in grouping bogus profiles together, we hypothesize that bisecting *k*-means clustering-based algorithm is robust against shilling attacks in non-private environments.

The paper is organized as follows. Section 2 discusses relevant literature and describes shilling attacks. We explain how bisecting *k*-means clustering-based CF operates on non-privately collected databases and discuss how shilling attacks can be implemented to modify its outputs in Section 3. Section 4 experimentally evaluates the robustness of the algorithm against shilling attacks in non-private environments. Finally, conclusions as a brief discussion and future research directions are presented in Section 5.

 ISBN: 978-1-61208-415-2

II. Related Work and Preliminary concepts

CF idea was first coined by the Tapestry system, which was utilized as a filtering tool for e-mails [19]. Contemporary CF technologies are integrated as recommender systems by online shopping amenities operating on preference data to produce personalized predictions [20]. Applications of CF schemes span from filtering e-mails [19] to Web service recommendations [21] and tag-based CF schemes [22].

With increasing popularity of CF systems, several attacking mechanisms arise to manipulate their outputs in favor of particular products. Dellarocas [23] inspires manipulation attacks to recommender systems, where some mechanisms are defined to avoid fraud in online reputation reporting systems. O'Mahony et al. [24] discuss vulnerabilities of automated prediction estimation process against manipulations. The authors describe the amount of information needed about the database to realize effective shilling attacks. Lam and Riedl [2][25] analyze cost of attacks and propose that there is a relation between privacy and the value of information. Several attacking methods are proposed in the literature like random, average, bandwagon, and segment attacks as push attacks [26]. Effectiveness of such attacks are investigated against memory- and model-based CF schemes [15]. Recently, Gunes et al. [27] surveyed about researches on shilling attacks and present attacks, detection methodologies, robust algorithms, and evaluation metrics.

Shilling attacks are generated by inserting fake (shilling) profiles into user-item databases. The general attack strategy is depicted in Fig. 1 [26], where I_S, I_F, and I_φ refers to selected, filler, and empty cells in the fake profile, respectively; and a unique item, i_t, is targeted. Selected items are chosen for characterizing an attack, filler items are chosen to prevent easy detection of fake profiles, and the target item is assigned either a high or a low rating value for push and nuke attacks, respectively. Shilling attacks can be used to increase the popularity of some targeted items or decrease their popularity. In order to push a prediction (increase the popularity of a target item), the target item is assigned a high rating. For decreasing the popularity of a target item, it is assigned a low rating.

Bisecting *k*-means clustering-based privacy-preserving recommendation algorithm is proposed to be easily scalable method and it produces predictions with high accuracy [17]. Notice that clustering-based CF schemes seem to be robust CF schemes without privacy concerns due to clustering nature. Hence, bisecting *k*-means clustering-based scheme might be appropriate proposal for being a robust algorithm. In our previous study [18], we investigated the robustness of privacy-preserving bisecting *k*-means clustering-based recommendation scheme against shilling attacks. In this study, we hypothesize that bisecting *k*-means clustering-based CF algorithm might be robust against shilling attacks due to its clustering nature in non-private environments, as well. Thus, we investigate its robustness against shilling attacks in non-private environments. We also provide comparisons between the proposed method and previously presented robust approaches in terms of obtained prediction shifts, algorithm interpretability, and model update costs. We focus on the robustness analysis of bisecting *k*-means clustering-based CF method against shilling attacks. As stated previously, bisecting *k*-means clustering-based recommendation algorithm is proposed as an accurate and scalable method. In this study, we want to show that it is also robust against shilling attacks in non-private environments.

III. A Robust Recommendation algortihm

Due to the reason that recommender systems are open for public usage and therefore vulnerable to manipulations, both non-private recommendation algorithms need to have robust mechanisms to estimate predictions. However, the state-of-the-art memory-based CF schemes are not resistant to such attacks and exposed to significant shifts in predicted values. In this section, we describe non-private bisecting *k*-means clustering-based recommendation scheme, designations of four push and two nuke attacking strategies against unmasked databases, and explain how the proposed algorithm is expected to perform in a robust manner.

A. Bisecting k-means Recommendation Algorithm

Bisecting *k*-means clustering-based recommendation estimation is first proposed by Bilge and Polat [17] in order to produce personalized recommendations over plain and disguised databases. In the proposed non-private scheme, the central server collects original user vectors and forms a user-item matrix $U_{n\times m}$, where *n* and *m* represent number of users and items, respectively. At the beginning, the server forms a binary decision tree off-line by utilizing bisecting *k*-means clustering algorithm on the database. Given the database and an optimal value of number of neighbors (*N*), *k*-means clustering is applied to divide the matrix *U* into two clusters at each level (hence, it is called bisecting) and cluster centers are indexed to be used as a forwarding tool for each corresponding level. If number of users in any cluster exceeds *N*, then such clusters are continued to be divided recursively into subsets via *k*-means clustering. Finally, a binary decision tree is obtained having indexed cluster centers as branch nodes and grouped neighbor users at leaf nodes. The tree, in general, continues growing in such a way so that if any leaf node population exceeds the stopping criterion, the server immediately bisects that leaf node to grow. Therefore, it is a continual process to update the decision tree, which saves the central server to form the binary decision tree each time a user included in the system. Such mechanism enhances system maintainability and reduces model generation costs.

An example binary decision tree produced by the algorithm is presented in Figure 2, where initially there are 150 users and the stopping criterion is determined as 20 users. At the beginning, the algorithm divides 150 users into two clusters having 73 ad 77 users and cluster centers are indexed at the root as C_1^L and C_1^R for the left and right subtrees, respectively. Such process continues recursively for each subtree and cluster centers are recorded to be used for forwarding purposes until the algorithm reaches leaf nodes containing at most 20 users.

When an active user (*a*) asks a prediction, instead of calculating similarities with all users, the server only forwards the active user according to her similarity to two cluster centers at each level. By doing so, the leaf node that the user belongs is determined through forwarding. While traversing, two similarity calculations are performed at each level, where higher similarity determines next hope (either right or left). Although depth of binary decision tree (*d*) is dependent on n, intuitively, it is much less than *n* in large recommender systems suffering from scalability. Therefore, after the tree is formed, at most $2 \times (d-1) + N$ similarity computations are performed

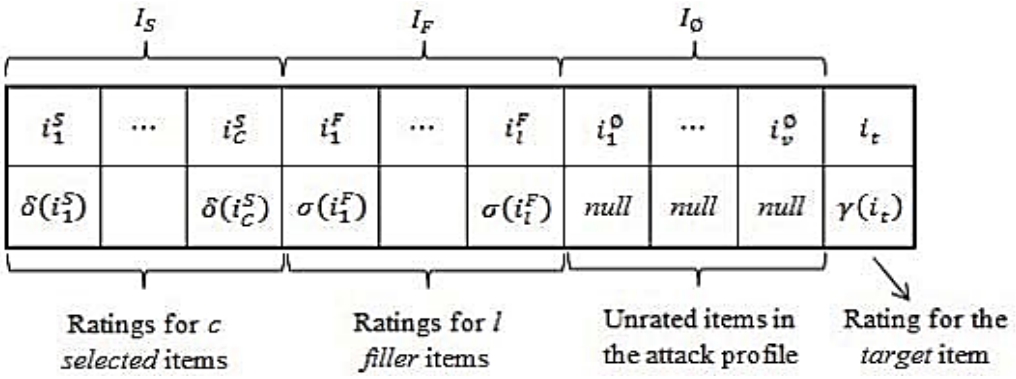


Figure 1. General form of an attack profile.

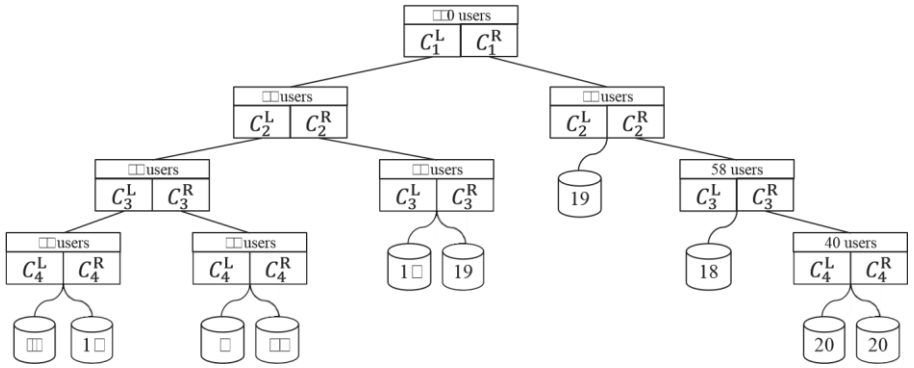


Figure 2. An example binary decision tree.

instead of *n* to form a neighborhood. Finally, the leaf node that the new user belongs is determined and all users in that corresponding node are regarded as neighbors. Then, a prediction is calculated as a weighted average of neighbors' ratings on target item as formulated in (1) and returned to *a* as a prediction.

$$p_{aq} = \overline{v_a} + \frac{\sum_{u \in N}(v_{uq} - \overline{v_u}) \times w_{au}}{\sum_{u \in N} w_{au}} \quad (1)$$

in which p_{aq} is the prediction for *a* on target item *q*, $\overline{v_a}$ and $\overline{v_u}$ are mean rating of *a*'s and *u*'s ratings, respectively, v_{uq} is the rating of *u* on item *q*, *N* is the set of neighbors, and w_{au} is the similarity weight between *a* and neighbor *u*.

B. Shilling Attack Strategies for Plain Databases

Shilling attacks have impacts on accuracy of the produced predictions. Attackers generate bogus profiles, assign their target items to maximum or minimum vote according to intends and insert them into the databases. Thus, they manipulate popularities of the target items in favor of themselves. Shilling attacks can be designed for pushing or nuking popularities of items. In order to perform manipulations, the attackers require low or high knowledge about the system. As part of their generic form depicted in Fig. 1, four push and two nuke attacks covered in this paper can be described as in the following [15]:

Random attack (RN). Random attack can be considered as a baseline *push* attack model, which requires quite minimal knowledge. Selected items set is empty and arbitrarily chosen filler items set is filled with random values drawn from a distribution with overall system mean and standard deviation for attacking non-private systems. The target item is assigned the maximum rating available in the system for non-private schemes.

Average attack (AV). Average attack is a more effective *push* attack model focusing on each item's individual mean rather than overall system's mean. Cost of this attack is related to the number of filler items in the attack profile because average votes of such items are required. Selected items set is empty and each arbitrarily chosen filler item is filled with a random value drawn from a distribution with corresponding item's ratings mean and standard deviation for attacking non-private systems. The target item is assigned the maximum rating available in the system for non-private schemes.

Bandwagon attack (BW). As a *push* attack model, bandwagon attack focuses on items that are attracting remarkable attention by many users to manipulate people who are prone to purchase such bestselling products. Selected items set consists of popular and densely-rated items having high averages. For attacking non-private systems, such selected items are given the maximum available rating, filler items are assigned random votes, and the target item is assigned the highest rating.

Segment attack (SG). Segment attack is designed as a *push* attack model for relatively robust item-based algorithms focusing on a subset (segment) of users who are likely to purchase certain kinds of products rather than attacking all users in the system. Selected items are chosen from high average items with a certain property (such as horror movies or jazz music). For non-private systems, such

 ISBN: 978-1-61208-415-2

selected items are assigned the maximum rating value, filler items are given the minimum rating value, and the target item is assigned the highest vote in order to push its popularity.

Reverse bandwagon attack (RBW). Reverse bandwagon *nuke* attack model is the inverted version of bandwagon push attack model. Selected items are chosen among unpopular items (having low means) rated by many users. For attacking non-private systems, such selected items are given the minimum available rating, filler items are assigned random votes, and the target item is assigned the lowest rating.

Love/hate attack (L/H). Love/hate attack is a very simple *nuke* attack model, which requires no knowledge about the system. For non-private systems, selected items set is empty and arbitrarily chosen filler items are assigned the highest available rating values while the target item is given the minimum vote.

C. Robustness Utility of the Recommendation Algorithm

Generally speaking, an attacker can attack any CF system by creating bogus profiles according to her intends as explained previously and sending them to the system. Thus, specifically, in order to attack non-private bisecting *k*-means clustering-based recommendation scheme, the attacker produces attack profiles and inserts them into the system. Since any CF scheme is vulnerable against shilling attacks, how well the scheme performs under such attacks is imperative for overall success. In other words, being robust against shilling attacks and/or able to detect bogus profiles are important.

In the previous studies [1][5][24], clustering-based CF schemes are shown to be successful in detecting fake profiles or bogus profiles. Arising from its utility of gathering similar data items together, clustering is utilized as a detection tool for shilling attacks in non-private schemes. O'Mahony et al. [24] utilize clustering as a neighborhood elimination method, where suspicious users are excluded from the system by clustering the database periodically to check if significant changes occur in memberships and cluster centers. If such significant changes occur, extreme profiles disturbing cluster centers are marked as malicious profiles. Bhaumik et al. [5] utilize *k*-means clustering with several classification attributes for attack detection. They show that shilling profiles show high resemblance to each other; therefore, when they are clustered, they tend to move together into the same and mostly small clusters. Especially, as initially determined number of clusters decrease, the likelihood of attack profiles gathering together increases.

Successful clustering-based schemes with respect to shilling attack detection inspire us to hypothesize that bisecting *k*-means clustering-based scheme can be proposed as a robust prediction algorithm. In addition to malicious profile detection, we hypothesize that clustering method can be utilized to offer robust recommendation algorithms. Relying on the results of [5], we hypothesize that elimination by clustering intuition works best for clustering into two groups to move shilling profiles together. In addition, applying such clustering repeatedly is supposed to eliminate all shilling profiles after some level of the produced binary decision tree. Therefore, we claim that malicious profiles substantially distinguishes from genuine ones after a particular level of the tree and it becomes very unlikely for any active user belonging to a leaf node consisting of shilling profiles. As a result, the proposed recommendation scheme is expected to perform robust against shilling attacks. To verify our hypothesis, we performed real data-based experiments as explained in the following section.

IV. EXPERIMENTAL EVALUATION

After explaining how shilling attacks can be implemented over non-private bisecting *k*-means recommendation algorithm, we conducted real data-based experiments to scrutinize the robustness of the scheme. We also investigated the effects of shilling attacks with respect to two control parameters. The control parameters, filler size and attack size, are defined for designing effective shilling attacks. Filler size parameter indicates the percentage of cells to be filled with fake ratings while creating the attack profiles. Attack size can be described as the pre-attack profile count proportional to the number of users in the database. We conducted various experiments for non-private bisecting *k*-means clustering-based CF scheme with varying values of the explained parameters.

A. Experimental Settings and Methodology

In the following experiments, publicly available MovieLens (ML) data set, which was collected by GroupLens [30], was utilized. It is the most widely used and well-known real collection for CF purposes. It holds 100K ratings from 943 users on 1,682 movies and the rating range allows 5-star discrete numeric values.

We used prediction shift metric in order to measure the prediction alterations due to the effects of shilling attacks. Prediction shift can be described as the average change in the prediction for the attacked item before and after the attack profiles are included.

During the experiments, we followed all-but-one experimentation methodology, which enables full utilization of the data set. This methodology considers one of the users as the active user *a* and the rest of the set as the training users at each iteration. The utilized attacks target two separate sets of 50 movies for push and nuke attacks. Those sets for push and nuke attacks were constructed selecting arbitrarily from different rating ranges to represent characteristics of the original data set. Since it is unreasonable to push a popular item with high ratings or similarly nuke an unpopular item, we principally selected items with low mean values to push and high means to nuke. Table I shows the statistics of 50 target movies for push and nuke attacks, where each value indicates how many of the movies fall into corresponding group.

In the experiments, all target items were attacked individually for all users in the system. Binary decision trees were constructed by omitting and including fake shilling profiles. Then, predictions were estimated based on the produced binary decision trees and prediction shift values were observed to show relative change on predicted values for different shilling attacks. The stopping criterion for building binary decision trees was set to 30. Although varying stopping criterion might alter obtained prediction shift values especially with varying attack sizes, we fixed such parameter due to page limitations and discuss algorithm's robustness performance relying on a constant stopping condition value. We exclusively presented the obtained results for push and nuke attacks in the following sections.

 ISBN: 978-1-61208-415-2

TABLE I. STATISTICS OF TARGETED MOVIES

Ratings	*Pushed IItems* 1–2	2–3	*Nuked Items* 3–4	4–5
1–50	30	15	12	18
51–150	—	3	5	6
151–250	—	1	2	3
250 and up	—	1	1	3

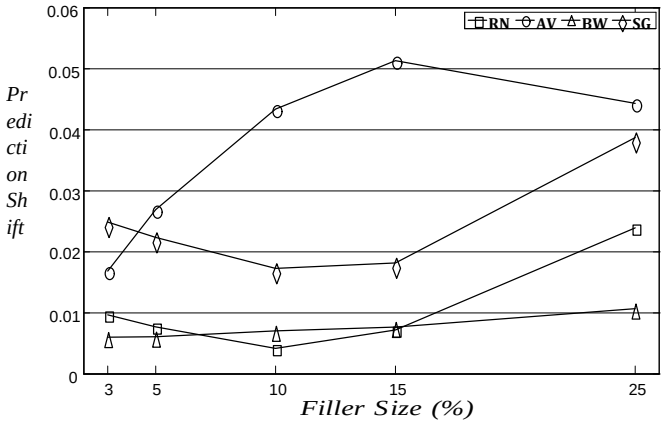


Figure 3. Prediction shift vs. filler size for push attacks.

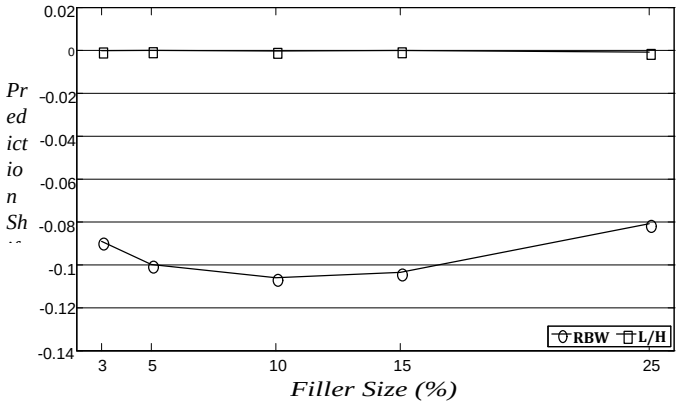


Figure 4. Prediction shift vs. filler size for nuke attacks.

B. *Robustness Analysis of Non-private Scheme*

1) Effects of filler size parameter: We first conducted experiments to show how varying filler size values affect the robustness of the non-private bisecting *k*-means clustering-based prediction scheme with respect to four push and two nuke attack models. Notice that filler size parameter indicates the number of fake votes for the filler items added to fill the attack profile; and thus, it is directly related to the success of the attack. To observe how varying filler size values affect robustness, we fixed attack size at 15% while we changed filler size from 3% to 25%. User-item matrix was attacked by four push and two nuke attack models. We estimated prediction shift values and displayed the overall averages for push and nuke attack models in Fig. 3 and Fig. 4, respectively.

As seen from Fig. 3, none of the four push attack models are able to achieve a significant prediction shift for varying filler size values. Generally speaking, with increasing filler size values, the effects usually become larger; however, increasing the value of filler size more is not feasible for the sake of detection of the attacks. The maximum prediction shift is observed for average attack when filler size is 15%. Compared to random and bandwagon attacks, average and segment attacks work better. However, their effects on the robustness of the scheme is still negligible because the maximum prediction shift is about 0.05 only. For bandwagon attack, changes in prediction shift values with increasing filler size values are very stable even if prediction shift values become larger. With increasing filler size values from 3% to 15%, there are notable changes in prediction shift values for average attack. As stated before, they are still insignificant assuming that the overall mean absolute error for the scheme is about 0.70. Therefore, we can conclude that bisecting *k*-means clustering-based prediction algorithm is robust against push attacks in non-private environments.

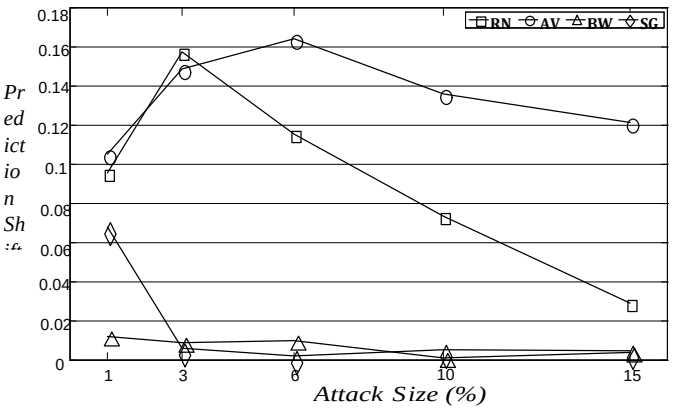


Figure 5. Prediction shift vs. attack size for push attacks.

The results in Fig. 4 show that nuke attack models are not effective against the non-private scheme with respect to varying filler size values. Changes in prediction shift values due to love/hate attack with increasing filler size values are insignificant. In other words, prediction shifts due to such attack are almost zero. Thus, love/hate attack is completely ineffective. Unlike love/hate attack, reverse bandwagon attack causes manipulations and it is more effective than love/hate attack. The maximum prediction shift value is about 0.1 for reverse bandwagon attack. Prediction shift values increase with increasing filler size values up to 10%, and then they decreases. However, such changes can be considered negligible due to the rating range. Hence, we can again conclude that bisecting k-means clustering-based prediction algorithm is resistant against nuke shilling attacks in non-private environments.

2) Effects of attack size parameter: We then performed various trials to show how varying attack size values affect the robustness of the non-private bisecting *k*-means clustering-based prediction scheme with respect to four push and two nuke attack models because in addition to filler size, attack size is another control parameter. Also note again that attack size determines the number of inserted attack profiles; thus, it is also vital in realizing significant manipulations. In order to evaluate the robustness of the non-private scheme with respect

to varying attack size values, we set filler size to 15% while we changed attack size from 1% to 15%. We again estimated prediction shift values and displayed the overall averages for push and nuke attack models in Fig. 5 and Fig. 6, respectively.

As seen from both figures, attack size is more effective than filler size parameter. The outcomes in Fig. 5 demonstrate that the most effective push attack in terms of attack size is average attack. The next most effective attack is random attack. Compared to both average and random attacks, segment and bandwagon attacks can be considered ineffective against the non-private method. Segment and bandwagon attacks cause stable changes in predictions with increasing attack size values. Almost all attack size values, prediction shifts for such attacks are very close to 0.01, which is negligible. Therefore, we can infer that our scheme is very robust against them and they do not significantly cause any manipulations. Although average and random attacks cause manipulations, the maximum shift is about 0.16 when the attack size is 6%. With increasing attack size values from 6% to 15%, prediction shift values for average and random attacks become smaller. The outcomes, in general, demonstrate that the non-private scheme is robust against push attacks in terms of varying attack size values.

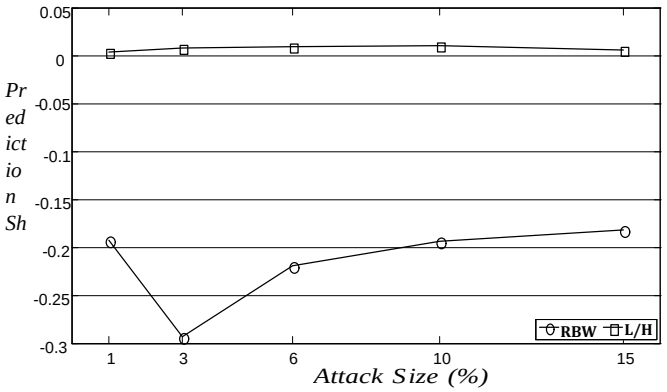


Figure 6. Prediction shift vs. attack size for nuke attacks.

Reverse bandwagon attack seems to be the most effective shilling attack, as seen from Fig. 6. When the attack size is 3%, prediction shift value reaches its maximum value, which is about 0.28. Other than 3% attack size value, predictions shift values are less than 0.20 for almost all other attack size values. Unlike reverse bandwagon attack, love/hate attack is much more ineffective. Although love/hate is used as a nuke attack and it is supposed to cause negative shifts, it causes negligible positive shifts. Moreover, changes in prediction shift values for varying attack size values for love/hate attack are stable and very close to zero. Thus, we can conclude that our scheme is very robust against love/hate attack.

C. Discussion

In addition to accuracy and scalability, robustness is also a critical requisite for recommendation algorithms. Due to its grouping nature, clustering has been used as a successful shilling attack detection method [1][5][6]. Thus, we hypothesized that bisecting *k*-means clustering-based algorithm can be proposed as a robust recommendation algorithm due to its clustering performance. We analyzed its robustness against six well-known shilling attacks (including both push and nuke attacks) in non-private environments. Bisecting *k*-means clustering-based scheme is robust in non-private environments, as shown by our real data-based trials. All of the push attacks that we scrutinize are ineffective against our scheme. Prediction shift values caused by such attacks are usually less than 0.05. In some cases, although prediction shift values reach at 0.16, they are still acceptable shifts compared to rating range. Average attack seems to be most effective push attack against our non-private method.

Like push attacks, nuke attacks can be considered ineffective against our scheme. Love/hate attack causes almost zero shifts in most of the cases. Therefore, it is not a good attack model to attack our non-private scheme. Unlike love/hate, reverse bandwagon is much more effective attack model against the non-private method. Prediction shift values due to reverse bandwagon attack reach 0.30 when attack size is set to 3%. Other than that case, prediction shifts caused by reverse bandwagon nuke attack are usually less than 0.20. Real data-based empirical outcomes demonstrate that our non-private recommendation method is robust against both push and nuke attacks. Out of six attack models, three of them (love/hate, bandwagon, and segment) are almost ineffective in many cases. Although reverse bandwagon, average, and random attacks seem to cause some prediction shifts, they are considered negligible due to the rating range.

In order to give an idea how robust our non-private scheme is, we compare it with the existing well-known recommendation algorithms with respect to robustness. According to study conducted by Mobasher et al. [28], average prediction shift values due to average attack are larger than 1.5 and 2.5 for *k*-means- and *k*-*nn*-based recommendation algorithms, respectively when attack size is 15% and filler size is 5%. For the same cases, average prediction shift values caused by segment attack are about 0.5 and 3.5 for *k*-means- and *k*-*nn*-based recommendation algorithms, respectively. Therefore, compared to them, our scheme is much more robust algorithm. Zhang et al. [29] show that prediction shift values are less than 0.003 for SVD-based prediction algorithm. Although the authors report that SVD-based scheme is a robust algorithm against shilling attacks and it is more robust than our scheme for reverse bandwagon, average, and random attacks, SVD-based model needs to be updated whenever a new user joins the system. Item-based recommendation algorithm is very susceptible against segment attack [15]. According to their empirical outcomes, average prediction shift caused by segment attack is larger than 0.9 when attack size is 15%. Similarly, bandwagon and average attacks cause more than 0.3 and 0.5 prediction shifts, respectively under the same cases. Therefore, our bisecting *k*-means clustering-based method performs better than item-based scheme with respect to shilling attacks.

V. Conclusion and Future Work

A prediction algorithm should handle various issues in order to become popular. Recommendation algorithms should provide accurate predictions, be scalable and robust, and so on. Thus, we investigated a formerly proposed accurate and scalable bisecting *k*-means clustering-based prediction algorithm's robustness against malicious shilling attacks in non-private environments. We first implemented four well-known push and two nuke attacks in non-private environments. We explained how such inserted attack profiles can affect the

recommendation scheme and why it is expected that the algorithm is robust against them. According to the obtained experimental results, the demonstrated push and nuke attack models are not able to significantly alter final predictions produced by the scheme. Thus, the algorithm is robust against shilling attacks in non-private environments. We scrutinized the effects of varying values of two control parameters like attack size and filler size. Although prediction shift values become larger as values of such parameters increase, prediction shifts are still acceptable. Our empirical results show that love/hate nuke attack is not effective against our scheme. So, even if attackers insert so many shilled profiles to our scheme, the scheme still produces accurate recommendations. Therefore, our scheme becomes more preferable than other recommendation schemes in terms of robustness in order to provide accurate predictions. Reverse bandwagon nuke attack is able to manipulate ratings; however, such manipulations are negligible. The non-private scheme performs better than item-based, *k-nn* clustering-based, and *k*-means clustering-based prediction schemes in terms of robustness against shilling attacks. Although singular value decomposition-based method is more robust than our non-private scheme, its complex model update process and model update requirement for each new user make it questionable.

It is known that clustering algorithms can be effective as a detection mechanism for shilling attacks. Hence, it warrants future work to utilize this algorithm as a detection tool of shilling profiles. Like segment attack, specific attack models can be designed as successful attacks.

ACKNOWLEDGEMENT

This work was supported by the Grant 111E218 from TUBITAK.

REFERENCES

[1] B. Mehta and T. Hofmann, “A survey of attack-resistant collaborative filtering algorithms,” IEEE Data Engineering Bulletin, vol. 31, no. 2, pp. 14–22, 2008.

[2] S. K. Lam and J. T. Riedl, “Shilling recommender systems for fun and profit,” Proc. 13th International Conference on World Wide Web, New York, NY, USA, pp. 393–402, 2004.

[3] J. Lang, M. Spear, and S. F. Wu, “Social manipulation of online recommender systems,” Lecture Notes in Computer Science, vol. 6430, pp. 125–139, 2010.

[4] B. Mobasher, R. D. Burke, R. Bhaumik, and C. A. Williams, “Effective attack models for shilling item-based collaborative filtering systems,” Proc. 2005 WebKDD Workshop, Chicago, IL, USA, 2005.

[5] R. Bhaumik, B. Mobasher, and R. D. Burke, “A clustering approach to unsupervised attack detection in collaborative recommender systems,” Proc. 7th IEEE International Conference on Data Mining, Las Vegas, NV, USA, pp. 181–187, 2011.

[6] R. D. Burke, B. Mobasher, C. A. Williams, and R. Bhaumik, “Classification features for attack detection in collaborative recommender systems,” Proc. 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, pp. 542–547, 2006.

[7] K. Choi and Y. Suh, “A new similarity function for selecting neighbors for each target item in collaborative filtering,” Knowledge-Based Systems, vol. 37, pp. 146–153, 2013.

[8] Z. Liang, X. Bo, and G. Jun, “A hybrid approach to collaborative filtering for overcoming data sparsity,” Proc. 9th International Conference on Signal Processing, Beijing, China, pp. 1595–1599, 2008.

[9] X. Luo, Y. Xia, and Q. Zhu, “Incremental collaborative filtering recommender based on regularized matrix factorization,” Knowledge-Based Systems, vol. 27, pp. 271–280, 2012.

[10] M. G. Vozalis, A. Markos, and K. G. Margaritis, “Collaborative filtering through SVD-based and hierarchical nonlinear PCA,” Lecture Notes in Computer Science, vol. 6352, pp. 395–400, 2010.

[11] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, “Eigentaste: A constant time collaborative filtering algorithm,” Information Retrieval, vol. 4, no. 2, pp. 133–151, 2001.

[12] S. Russell and V. Yoon, “Applications of wavelet data reduction in a recommender system,” Expert Systems with Applications, vol. 34, no. 4, pp. 2316–2325, 2008.

[13] A. Bilge and H. Polat, “A comparison of clustering-based privacy-preserving collaborative filtering schemes,” Applied Soft Computing, vol. 13, no. 5, pp. 2478–2489, 2013.

[14] O. Georgiou and N. Tsapatsoulis, “Improving the scalability of recommender systems by clustering using genetic algorithms,” Lecture Notes in Computer Science, vol. 6352, pp. 442–449, 2010.

[15] B. Mobasher, R. D. Burke, R. Bhaumik, and C. A. Williams, “Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness,” ACM Transactions on Internet Technology, vol. 7, no. 4, pp. 23–60, 2007.

[16] M. G. Vozalis, A. Markos, and K. G. Margaritis, “On the performance of SVD-based algorithms for collaborative filtering,” Proc. 4th Balkan Conference in Informatics, Thessaloniki, Greece, pp. 245–250, 2009.

[17] A. Bilge and H. Polat, “A scalable privacy-preserving recommendation scheme via bisecting *k*-means clustering,” Information Processing & Management, vol. 49, no. 4, pp. 912–927, 2013.

[18] A. Bilge, I. Gunes, and H. Polat, “A robust privacy-preserving recommendation algorithm,” Proc. 2nd Asian Conference on Information Systems, Phuket, Thailand, 2013.

[19] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, “Using collaborative filtering to weave an information Tapestry,” Communications of the ACM, vol. 35, no. 12, pp. 61–70, 1992.

[20] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,” ACM Transactions on Information Systems, vol. 22, no. 1, pp. 5–53, 2004.

[21] J. Cao, Z. Wu, Y. Wang, and Y. Zhuang, “Hybrid collaborative filtering algorithm for bidirectional Web service recommendation,” Knowledge and Information Systems, vol. 36, no. 3, pp. 607–627, 2013.

[22] H. Movahedian and M. R. Khayyambashi, “A tag-based recommender system using rule-based collaborative profile enrichment,” Intelligent Data Analysis, vol. 18, no. 5, pp. 953–972, 2014.

[23] C. Dellarocas, “Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior,” Proc. 2nd ACM Conference on Electronic Commerce, Minneapolis, MN, USA, pp. 150–157, 2000.

[24] M. P. O’Mahony, N. J. Hurley, and G. C. M. Silvestre, “Collaborative filtering - safe and sound?” Lecture Notes in Computer Science, vol. 2871, pp. 506–510, 2003.

[25] S. K. Lam and J. T. Riedl, “Privacy, shilling, and the value of information in recommender systems,” Proc. User Modeling Workshop on Privacy-Enhanced Personalization, Edinburgh, UK, pp. 85–92, 2005.

[26] B. Mobasher, R. D. Burke, C. A. Williams, and R. Bhaumik, “Analysis and detection of segment-focused attacks against collaborative recommendation,” Lecture Notes in Computer Science, vol. 4198, pp. 96–118, 2006.

[27] I. Gunes, C. Kaleli, A. Bilge, and H. Polat, “Shilling attacks against recommender systems: A comprehensive survey,” Artificial Intelligence Review, vol. 42, no. 4, pp. 767–799, 2014.

[28] B. Mobasher, R. Burke, and J. J. Sandvig, “Model-based collaborative filtering as a defense against profile injection attacks,” Proc. 21st National Conference on Artificial Intelligence - Volume 2, Boston, MA, USA, pp. 1388–1393, 2006.

[29] S. Zhang, Y. Ouyang, J. Ford, and F. Makedon, “Analysis of a low-dimensional linear model under recommendation attacks,” Proc. 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, WA, USA, pp. 517–524, 2006.

[30] “Non-commercial, personalized movie recommendations” MovieLens. Web. 11 Apr. 2015.

 ISBN: 978-1-61208-415-2

Improving Relevance Effectiveness in Data Leakage Detection Using Feature Selection

Adrienn Skrop
Department of Computer Science and Systems Technology
University of Pannonia
Veszprém, Hungary
e-mail: skrop@dcs.uni-pannon.hu

***Abstract*— Data leakage is an uncontrolled or unauthorized transmission of classified information to the outside. Many software solutions were developed to provide data protection. However, none of them can provide absolute protection. The purpose of the research is to design and implement DATALEAK, a data leakage detection system based on information retrieval models and methods. In this paper, a feature selection based information retrieval model is proposed to improve relevance effectiveness of DATALEAK. The paper focuses on dimensionality reduction, where semantic matching of documents is performed in the reduced form of the vector space model.**

Keywords—data leakage; vector space model; feature selection.

I. INTRODUCTION

Data leakage is an event in which classified information has been viewed, stolen or used by somebody who is not authorized to do so. Data leak prevention methods and systems helps ensure that confidential data remain safe and secure [4][6][8]. However, data leakage detection systems cannot provide absolute protection. On the one hand, more than 40 per cent of data breaches are due to insider negligence [3]. On the other hand, 80 to 90 percent of an organization's data is unstructured. Unlike application oriented data, which is usually well structured and has means of protection, unstructured data is loose, out of control and hard to protect.

In [1][2], DATALEAK, a semantic information-retrieval (IR) based application is presented to address the problem of Web data leakage detection. The system uses a vector space model (VSM) based representation to compare documents. Having a high dimensional VSM, it is impractical to calculate similarity measure. In this paper, we propose an approach to increase effectiveness by reducing the dimensionality of the system.

In Section II, the DATALEAK system is presented briefly. Section III presents the feature selection method that is planned to be implemented in the system. Section IV concludes the paper.

II. THE DATALEAK SYSTEM

The goal of the DATALEAK system is to monitor the Web and collect Web documents according to users' preferences. The collected Web documents are compared with user's confidential documents. If a document turns up on the Web that is semantically similar to confidential user documents the system indicates potential data leakage. The DATALEAK system is composed of the following modules. The Search module is responsible for discovering Web pages that might indicate data leakage. The search module is implemented as a conventional keyword-based metasearch engine. The engine uses the hit list of Google and Bing. The Text mining module is responsible for the automated processing of Web pages that were identified by the Search module. The Text mining module converts Web documents into their appropriate mathematical representation. During automated processing relevant keywords or index terms are extracted from Web documents. Using the extracted keywords Web documents can be represented in a vector space. The document collection contains confidential user documents. The Cryptographic Module is responsible for preparing an encrypted version of these documents. It works similarly as the Text mining module. The input can be any user document. The output is a set of keywords. The keywords are used to create a mathematical representation of user documents. The Scoring module matches the mathematical representations of Web documents and confidential user documents. A number of mathematical models can be used to represent documents and to calculate similarity. In the next section, a reduced dimensional vector space model is presented.

III. DIMENSIONALITY REDUCTION BY FEATURE SELECTION

Usually, the similarity of documents is determined using a repetition-based hard similarity metric. This approach ignores all potential semantic correlations between different words. In DATALEAK, not the pure content, but the meaning of Web documents and user documents are compared.

Given a search query, the retrieved Web documents and confidential user documents, the Scoring module computes a relevance score that measures the similarity between these documents. The scoring module uses the VSM representations of documents. In VSM, documents are represented by a vector in an n dimensional vector space, where n is the number of keywords or index terms [7].

The VSM can be created in three steps. The first step is indexing where keywords are extracted from the documents. Many of the words in a document do not describe the content. These words are called stop words, e.g. the, like, is etc...

 ISBN: 978-1-61208-415-2

By using automatic document indexing these non-significant, usually high frequency words are removed from the document, so the document will only be represented by content bearing words, i.e. index terms.

The second step is the weighting of the indexed terms. A term can be assigned a weight that expresses its importance for a particular document. A common weighting scheme for terms within a document is to use the frequency of occurrence, called by term frequency [9]. The term frequency can be used as a content descriptor for the documents and is generally used as the basis of a weighted document vector [10].

The last step is to compare documents according to some similarity measure. The most popular similarity measure is the cosine coefficient, which measures the angle between the document vector and the query vector.

In the data leakage detection system, the basic VSM representation is as follows.

During indexing, a set of keywords $C = \{c_1, ..., c_i\}$ are extracted from confidential documents and another set of keywords $W = \{w_1, ..., w_j\}$ are extracted from Web documents. Web documents and user confidential documents are represented using the VSM over $C \cup W$ as follows. Given a finite set $T = C \cup W$ of index terms $T = \{t_1, ... ,t_n\}$ any Web document D_j is assigned a vector $\mathbf{v}_j$ of finite real numbers:

$$\mathbf{v}_j = (w_{ij})_{i=1,...,n} = (w_{1j}, ..., w_{ij}, ..., w_{nj}) \quad (1)$$

Confidential user documents U_k also have to be represented as a vector v_k of finite real numbers, as follows:

$$\mathbf{v}_k = (w_{ik})_{i=1,...,n} = (w_{1k}, \ldots, w_{ik}, \ldots, w_{nk}) \quad (2)$$

A Web document D_j is represented to a user having confidential document U_k if they are similar enough, i.e., a similarity measure S_{jk} between the Web document vector $\mathbf{v}_j$ and the confidential user document vector $\mathbf{v}_k$ is over some threshold K. The threshold K should be chosen to represent a required level of lower bound for the similarity of two documents.

The disadvantage of this representation is that, thanks to the Web documents' potential diversity, the dimensionality of the vector space can be fairly high causing costly computation of the similarity measure. A solution to this problem can be to reduce the size of the vector space. Feature subset selection is a technique that is used in supervised and unsupervised classification or regression problems for reducing the attribute space of a feature set. The purpose of feature selection is to identify significant index terms and eliminate irrelevant ones [5]. In this paper, feature selection is proposed to be used in VSM to reduce the dimensionality of the vector space. In order to emphasis the importance of user documents, keywords are extracted only form user documents. These keywords will be used to form the dimensions of the vector space. User documents and Web documents are represented over this limited VSM. Besides reducing the dimensionality of the vector space, feature selection also might contribute to improve precision, i.e., to ensure that the model becomes specialized enough to represent confidential user documents. Fig. 1 visually represents the proposed approach as opposed to the previously presented vector space based matching. We consider that the automatic indexing process has associated a set of keywords $C = \{c_1, ..., c_i\}$ to confidential user documents. The user is allowed to modify the set of keywords by adding new, content bearing keywords $E = \{e_1, ..., e_j\}$, e.g. synonyms; and removing improper keywords $R = \{r_1, ..., r_k\}$.

The result is a VSM over $C + E - R$ as follows. Given a finite set $T' = C + E - R$ of new index terms $T' = \{t'_1, ... ,t'_m\}$, $m < n$, any confidential user document U_k is assigned a vector $\mathbf{v}'_k$ as follows:

$$\mathbf{v}'_k = (w_{ik})_{i=1,...,m} = (w_{1k}, \ldots, w_{ik}, \ldots, w_{mk}) \quad (3)$$

Any Web document D_j is assigned a vector v_j over this new reduced dimensional VSM as follows:

$$\mathbf{v}'_j = (w_{ij})_{i=1,...,m} = (w_{1j}, ..., w_{ij}, ..., w_{mj}) \quad (4)$$

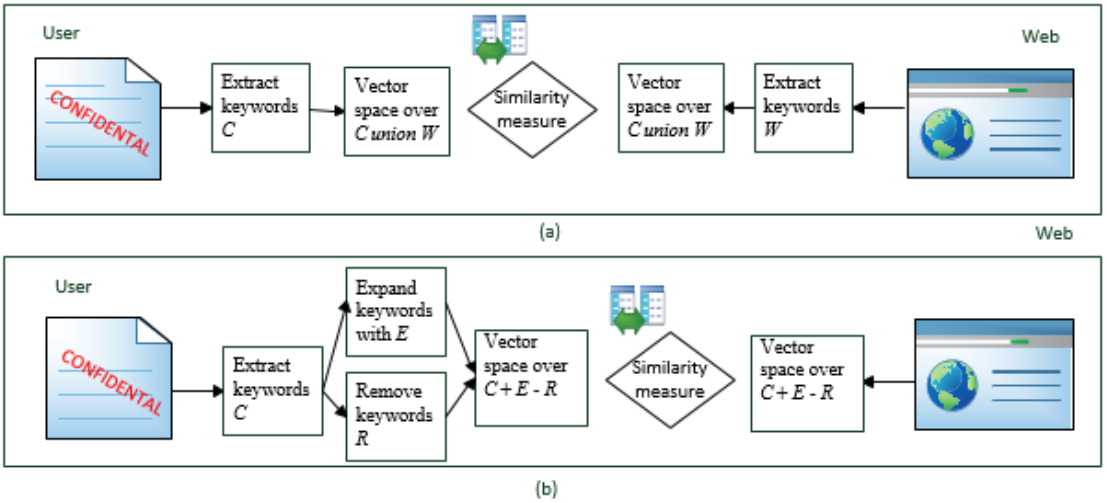


Figure 1. Block diagram showing (a) the vector space based matching, as opposed to (b) the reduced dimensionality based matching problem.

A Web document D_j is represented to a user having confidential document U_k if they are similar enough, i.e., a similarity measure S_{jk} between the Web document vector $\boldsymbol{v}'_j$ and the confidential user document vector $\boldsymbol{v}'_k$ is over some threshold K, i.e.,

$$S_{jk} = s(\boldsymbol{v}'_j, \boldsymbol{v}'_k) > K \quad (5)$$

The expansion of the vector space with new keywords is similar to query expansion. The goal is to improve effectiveness by matching related terms. Instead of adding new keywords manually, a variety of automatic or semi-automatic expansion techniques can be used [11]. Semi-automatic techniques require user interaction to select best expansion terms. In this application, the combination of two techniques is considered to be used. One technique is to use general or domain specific term taxonomies, e.g. WordNet, to determine a set of semantically similar keywords E_1 (e.g. synonyms and hyponyms). Another technique is relevance feedback, which relies on user interaction to identify relevant documents. Having the hit list produced by the Search module, the user indicates which documents are similar enough (relevant) and which documents are non-relevant. Only relevant documents are sent to Text mining module to extract a set of keywords E_2. Finally, the co-occurring keywords, i.e. $E = E_1 \cup E_2$ are selected to be added to the VSM.

IV. CONCLUSION

In this paper, we introduced a vector space based matching approach to address the problem of data leakage detection. The idea is to reduce the dimensionality of the vector space and improve relevance effectiveness by feature selection. Feature selection is based on confidential user documents in order to achieve better precision. Semantic matching of Web documents is performed against confidential documents in the reduced form of the vector space model to reduce complexity.

ACKNOWLEDGMENT

This research has been supported by the European Union and Hungary and co-financed by the European Social Fund through the project TÁMOP-4.2.2.C-11/1/KONV-2012-0004 - National Research Center for Development and Market Introduction of Advanced Information and Communication Technologies.

REFERENCES

[1] A. Skrop, "Data Leakage Detection Using Information Retrieval Methods," in: Schmidt, A., Yarali, A. (eds.). *IMMM 2014*, The Fourth International Conference on Advances in Information Mining and Management. IARIA. Paris, France, July 20-24. 2014. pp. 74-78. ISBN: 978-1-61208-364-3.

[2] A. Skrop, "DATALEAK: Data Leakage Detection System," MACRo2015, The 5th International Conference on Recent Achievements in Mechatronics, Automation, Computer Sciences and Robotics. Tirgu Mures, Romania, March 6-7, 2015, pp. 115-126. ISSN: 2247-0948.

[3] D. S. Wall, "Organizational security and the insider threat: Malicious, negligent and well-meaning insiders," Technical report, Symantec, 2011.

[4] E. Gessiou, Q. H. Vu, and S. Ioannidis, "IRILD: an Information Retrieval based method for Information Leak Detection," in Proceedings of European Conference on Computer Network Defense, 2011, pp. 33–40, IEEE.

[5] I. Guyon and E. André, "An introduction to variable and feature selection," The Journal of Machine Learning Research" vol(3), 2003, pp. 1157-1182.

[6] P. Papadimitriou and H. Garcia-Molina, "Data leakage detection," Knowledge and Data Engineering, IEEE Transactions on, vol. 23(1), 2011, pp. 51-63.

[7] R. Baeza-Yates and B. Ribeiro-Neto, Modern information retrieval: The Concepts and Technology behind Search (2nd Edition). ACM Press Books, Addison-Wesley Professional, 2011, ISBN: 0321416910.

[8] Y. Liu, C. Corbett, K. Chiang, R. Archibald, B. Mukherjee, and D. Ghosal, "SIDD: A framework for detecting sensitive data exfiltration by an insider attack," In System Sciences, 2009, HICSS'09, pp. 1-10, IEEE.

[9] H. P. Luhn, "The Automatic Creation of Literature Abstracts", IBM Journal of Research and Development 2 (2), 1958, pp. 159-165 and 317.

[10] G. Salton, Gerard and C. Buckley, "Term-weighting approaches in automatic text retrieval," Information processing & management, 24.5, 1988, pp. 513-523.

[11] E. N. Efthimiadis, "Query expansion," Annual review of informasion systems and technology (ARIST), vol. 31. 1996, pp.121-187.

 ISBN: 978-1-61208-415-2

Analyzing and Improving Educational Process Models using Process Mining Techniques

Awatef Hicheur Cairns[1], Billel Gueni[1], Joseph Assu[1], Christian Joubert[1], Nasser Khelifa[2]

[1]ALTRAN Research, [2]ALTRAN Institute
Vélizy-Villacoublay, France
e-mails: {awatef.hicheurcairns, billel.gueni, assu.joseph, christian.joubert, nasser.khelifa}@altran.com

***Abstract*— Educational process mining is an emerging field in the educational data mining discipline, concerned with discovering, analyzing, and improving educational processes as a whole, based on information hidden in educational datasets and event logs. In this paper, we demonstrate the applicability of process mining techniques, implemented in the ProM framework, to monitor and analyze educational processes in the field of professional trainings. Furthermore, we extended the discovered training processes with performance characteristics and decision rules using performance and decision mining techniques.**

Keywords- Educational Process Mining; Conformance Ckecking; Decision Minin; Performance Analysis; ProM.

I. INTRODUCTION

Given the ever changing needs of the job markets, education and training centers are increasingly held accountable for student success. Therefore, education and training centers have to focus on ways to streamline their offers and educational processes in order to achieve the highest level of quality in curriculum contents and managerial decisions. To respond to these requirements, education and training centers promote more flexible and personalized curriculums where students are free to choose the skills they want to develop, the way they want to learn and the time they want to spend. This tendency is reinforced by the emergence of "e-learning", which represents an increasing proportion of the in-company trainings, while addressing ever wider populations [10]. Educational systems support a large volume of data, coming from multiple sources and stored in various formats and at different granularity levels. These data can be analyzed from various levels and perspectives, showing different aspects of educational processes from the view points of the students, the educators or the directors of education centers [10, 11]. Recently, Educational Process mining has emerged as a promising and active research field in Educational Data Mining [11], dedicated to extracting process related-knowledge from educational datasets. The basic idea of process mining [2] is to discover, monitor and improve real processes by extracting knowledge from event logs (recorded by an information system). In this paper, we focus on educational process monitoring and improvement using performance analysis, conformance checking and process model extension techniques. We take as a case study a professional training dataset of a consulting company involved in the training of professionals. This work is motivated by the fact that training mangers aim to gain more insight in employees' training paths and motivation so they can offer more personalized training courses, according to the job market needs. Therefore, our aim is to (1) analyze training processes and their conformance with established curriculum constraints, educators' hypothesis and prerequisites and (2) to enhance training process models with performance indicators such as execution time, bottlenecks and decision points. We use the process mining tool ProM as an execution framework in our study.

The remainder of this paper is organized as follows: Section II reviews related works. Section III summarizes process mining techniques. In Section IV, we present our motivating example and we show the use of ProM's plugins for the analysis and the enhancement of training process models. Finally, Section V concludes the paper.

II. RELATED WORKS

In [8], process model discovery and analysis techniques, were used to investigate the students' behavior during online multiple choice examinations. In [14], the authors use process mining techniques to analyze a collaborative writing process and how the process correlates to the quality of the produced document. In [16], the authors proposed a technique relying on a set of predefined pattern templates to extract pattern-driven education models from students' examination traces (i.e., by searching for local patterns and their further assembling into a global model). In [16, 17], the authors developed the first software prototype for academic curriculum mining, built on the ProM framework. This tool monitors the flow of curriculums in real-time and return warnings to students if prerequisites are not satisfied. Two clustering approaches were proposed in [4], grouping students relying on their obtained marks and their interaction with the Moodle's course. Performance analysis techniques were used to detect bottlenecks in students' registration processes in [3]. Finally, in our previous work [5], we showed how social mining techniques can be used to

 ISBN: 978-1-61208-415-2

examine and assess interactions between training providers and courses. We also proposed a two-step clustering approach for partitioning training processes depending on an employability indicator. In comparison with our pervious works, we focus in this paper mainly on educational process monitoring, using performance and conformance analysis techniques. We also studied the applicability of process model extension techniques such as decision mining which have never used in the context of educational process mining. Our goal is to show the advantages of these techniques for the analysis of professional training processes in particular and also their limitations regarding the size of the analyzed event logs. This study help us understand which are the most relevant process mining techniques to integrate in our interactive and distributed platform, tailored for educational process discovery and analysis, and which is currently under construction.

III. EDUCATIONAL PROCESS MINING

Process mining focuses on the development of a set of intelligent tools and techniques aimed at extracting process-related knowledge from event logs [2]. An *event log* corresponds to a set of process *instances* (i.e., traces) following a business process. Each recorded *event* refers to an *activity* and is related to a particular process instance. An event can have a *timestamp* and a *performer* (i.e., a person or a device executing or initiating an activity). Typical examples of event logs in education may include students' registration procedures and attended courses, student's examination traces, use of pedagogical resources and activity logs in e-learning environments. The three major types of process mining techniques are: (1) *discovery*, (2) *conformance* and (3) *extension*. *Process model discovery* takes an event log and produces a complete process model able to reproduce the behaviour observed in this log. *Conformance checking* aims at monitoring deviations between observed behaviours in event logs and normative process models [12]. *Compliance checking* aims at measuring the adherence of event logs with predefined business rules or Quality of Service (QoS) definitions [1]. *Process model extension* aims to improve a given process model based on information (e.g., time, performance, case attributes, decision rules, etc.) extracted from an event log related to the same process. The ProM Framework is the most complete and powerful process mining tool, with an extendable pluggable architecture, aimed at process discovery and analysis from all perspectives [18]. ProM supports a wide range of techniques for process discovery, conformance analysis and model extension. In practice, however, ProM presents certain issues of flexibility and scalability, which limit its effectiveness in handling large logs from complex industrial applications [9].

IV. CASE STUDY: AUDITING TRAINING PROCESSES USING PROCESS MINING TECHNIQUES

Our motivating example is based on real-world professional training databases from a worldwide consulting company. This company has around 6 000 employees that are free to choose different training courses aligned with their profiles, during their careers. These training courses are provided by internal or external training organizations. The data collected for analysis reports all the 16 260 training courses followed by 3440 employees, during the last three years, performed by 494 training organisations. This data includes the employees' profiles (identifier, function, and number of years of service), their careers (i.e., the jobs/missions they did) and their training paths.

TABLE I. EXAMPLE OF AN EDUCATIONAL EVENT LOG

Matricul	Profil	Training_Id	Training_Label	Training_Orga_Id	StartDate	EndDate
7	CONSULTANT	Tr 850	EXCEL ELEARNING	Org 13	11/07/2011	31/12/2011
8	CONSULTANT	Tr 769	QF TEST	Org 135	26/04/2011	28/04/2011
9	CONSULTANT	Tr 252	INTERCULTURAL WORKING RELATONS : INDIA	Org 135	01/07/2011	01/07/2011
10	CONSULTANT	Tr 260	SELENIUM	Org 135	25/10/2011	26/10/2011
11	CONSULTANT	Tr 812	UML FUNCTIONAL ANALYSIS	Org 135	24/10/2011	27/10/2011
12	CONSULTANT	Tr 774	DESIGN PATTERNS AND APPLICATION C++	Org 135	08/12/2011	09/12/2011
13	CONSULTANT	Tr 1923	SQL BASIC	Org 135	03/04/2012	05/04/2012
14	CONSULTANT	Tr 813	C++ ADVANCED	Org 135	04/04/2012	06/04/2012
15	CONSULTANT	Tr 2014	XML BASIC AND XPATH	Org 135	10/04/2012	11/04/2012
14	CONSULTANT	Tr 1282	DESIGN PATTERNS AND APPLICATION IN C++	Org 135	13/09/2012	14/09/2012
...	....		...		.....	

A. Dotted Chart Analysis

The *dotted chart* shows the spread of events over time by plotting a dot for each event in an event log thus allowing to gain some insight in the underlying process, its performance and some interesting patterns [15]. The chart has two orthogonal dimensions: time and component types. The time is measured along the horizontal axis of the chart. The component types (e.g., instance, originator, task, event type, etc.) are shown along the vertical axis. Figure 1 illustrates the output of the dot chart analysis (implemented in ProM 6.4 as a plugin) of the training log example using process instances as component type. In this chart, every row corresponds to a particular case of the training process, i.e., all the trainings followed by one employee during the last three years.

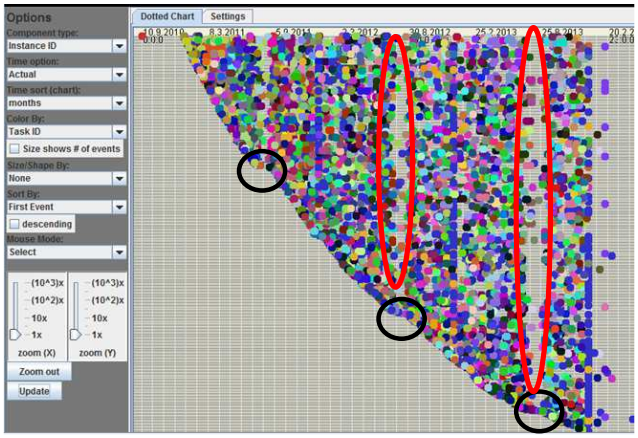


Figure 1. Dotted chart showing all events of the training log

Each training course is represented by a dot. All the instances (one per trainee) are sorted by the first events of trainings i.e., trainings are sorted by the first date of their occurrence. Figure 1 shows that each year, there are few training courses scheduled around the last three months (circled in black), probably because the entire budget is consumed around the end of November. Also, almost no training is scheduled during the summer (circled in red).

B. Trace Alignement

Let us note that in a dotted chart, common patterns among different cases are not clearly visible. For analysing common patterns between the training paths of the company's employees of our case study, we follow the technique presented in [6] and implemented as plugins in ProM 6, where the first step is based on the *`Guide Tree Miner'* plugin and the second steps are handled by the *`Trace Alignment with Guide Tree'* plugin (both implemented in ProM 6.4). The guide tree miner plugin requires an event log as input. This plugin implements the agglomerative hierarchical clustering algorithm for the generation of the guide tree and *k* event logs, one for each cluster. The *`Trace Alignment with Guide Tree'* plugin takes as input a guide tree generated by the guide tree miner plugin and generate an alignment of traces for each cluster. In our example, we have split the training event log into sixteen clusters. Figure 2 depicts the trace alignment for one of the 16 clusters where every row corresponds to a process instance and time increases from left to right. The horizontal position is based on logical time rather than real timestamps. When two rows have the same activity name in the same column, the corresponding events are aligned. The left panel depicts the employee identifier (i.e., process instance identifier) and identifiers with a grey background indicate traces that have identical duplicates. In Figure 2, there are 8 employees who follow the same training path that the employee with the identifier 29784. Also, the traces are sorted based on the training d0 (i.e., Internal Tool Delivriz). Consequently, training traces with d0 in column 3 have the first priority in the ordering.

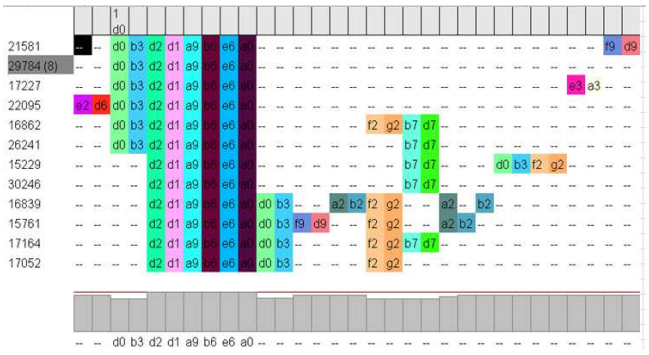


Figure 2. Trace alignment of traces in the training event log example for one of the clusters.

The bottom panel depicts the consensus sequence for the alignment (i.e., the major activity in each column and can be considered as a back-bone sequence for the process). The consensus sequence for this example d0b3d2d1a9b6e6a0 corresponds to a well conserved training pattern "Internal Tool Delivriz, Project Management Module 1, Project Management Module 2 and Project Management Module 3". Figure 2 shows also that the training "Internal Tool Delivriz" (encoded b3) is a concurrent activity. Concurrent activity manifests in mutually exclusive traces across different columns in the alignment. Trace alignment allows us to see such similarities between training traces inducing interesting training patterns. Given the great heterogeneity in training traces, only few clusters produced by the guide tree miner allow us to obtain a good trace alignment. In fact, the challenge of trace alignment is to find an alignment that is as simple and informative as possible when we analyse event log with such heterogeneity as the ones encountered in the education domain. To achieve this, some trivial pre-processing techniques such as the filtering of traces based on their length may help produce a better alignment.

C. Performance Analysis

Performance analysis with Petri net plugin of ProM can extract the Key Performance Indicators from an event log, summarizing them in an intuitive way, and graphically present them on a Petri net describing the process under consideration. When calculating the different performance metrics, all process instances in the input event log are replayed in the input Petri net. We apply this plugin on our training courses dataset in order (1) to predict learners' training path duration, (2) to predict necessary resources (i.e., trainers) at any moment within the training process and (2) to improve the training process by analysing the bottlenecks. Identifying bottlenecks in training paths can be used to better schedule training sessions to reduce the waiting time of trainees. Let us note that given the great heterogeneity in training traces, we can't use the *performance analysis* plugin on our complete training event log [9]. To get past this limitation, we apply the performance analysis plugin on only a subset of the training dataset example containing learning paths with only the six most frequent trainings. The result of the analysis is partly shown in Figure 3 displaying the bottlenecks (red coloured places) and the routing probabilities on each arc. As shown, 6% percent of the employees chose to take the training management project-M1 while 10% followed the training Information Technology Infrastructure Library (ITIL) Foundation. The average waiting time between two successive trainings is 45 days and the maximal waiting time is one year. Moreover, the average duration of a training path is 78 days while the longest training path lasted more than one year.

 ISBN: 978-1-61208-415-2

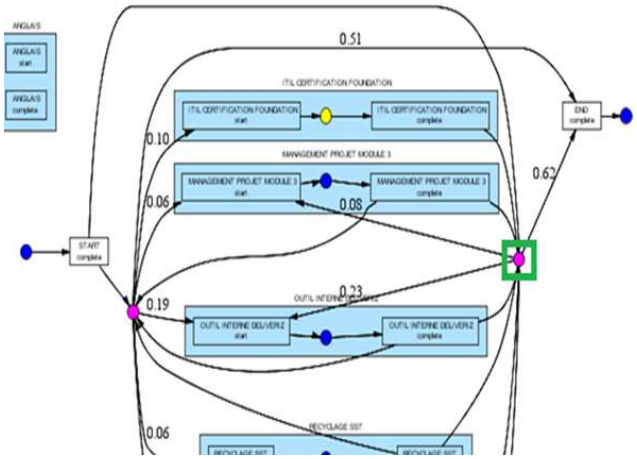


Figure 3. Results of applying the Performance analysis with Petri net plugin

D. *Conformance Analysis*

In what follows, we propose to use ProM's *Conformance Checker* and *LTL Checker* plugins in a training tailored procedure (see Figure 4) to check whether training paths, as they are really followed by employees, are conform to established constraints in the training curricular.

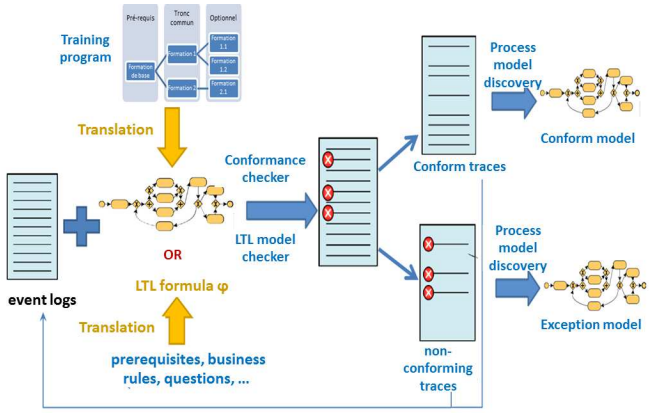


Figure 4. The proposed procedure for conformance analysis of training paths

The conforming and non-conforming traces produced by these two kinds of analysis can be used to extract conform training process models and exception training process models underlying these traces (respectively).

1) *Linear Temporal Logic (LTL) Analysis*

Training advisors and directors of training organisms often need to check (off-line or on-line) whether trainees' paths conform to established career paths, trainings' prerequisites or business rules. For this purpose, we use the *LTL Checker* plugin of ProM that allows us to check whether an event log satisfies a given set of properties expressed in terms of LTL logic [1]. There is a set of predefined formulas in the LTL model checker plugin of ProM. It is also possible to tailor the LTL checker plugin to express specific types of constraints encountered in the educational domain. All these properties can be easily coded using the LTL language of the plugin and imported as a LTL file into the user interface. In what follows, we want to check if the rule "*Project Management-Module-1* training must be taken before a *Project Management-Module* 3 can be taken" was always respected (prerequisite check). We define this property as an LTL formula as follows:

```
formula c2_is_a_prerequisite_of_c1
c1: ate.WorkflowModelElement,
c2: ate.WorkflowModelElement) :=
{<h2> Is the training C2 a prerequisite for the
training C1? </h2>}
(<>(activity==c2)/\(activity!=c2 _U
activity==c1));
```

Fig 5 shows the result displayed when this property is checked.

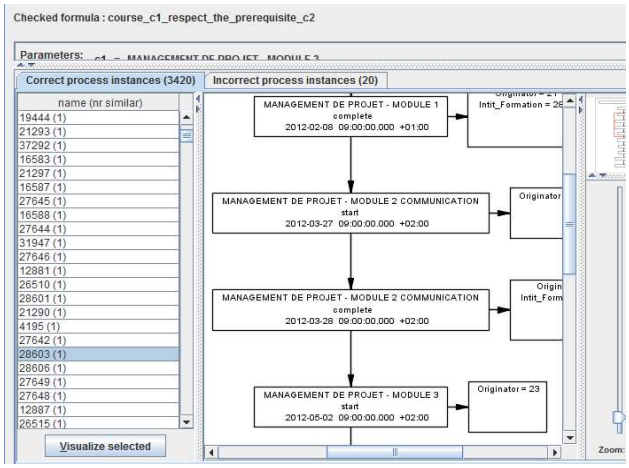


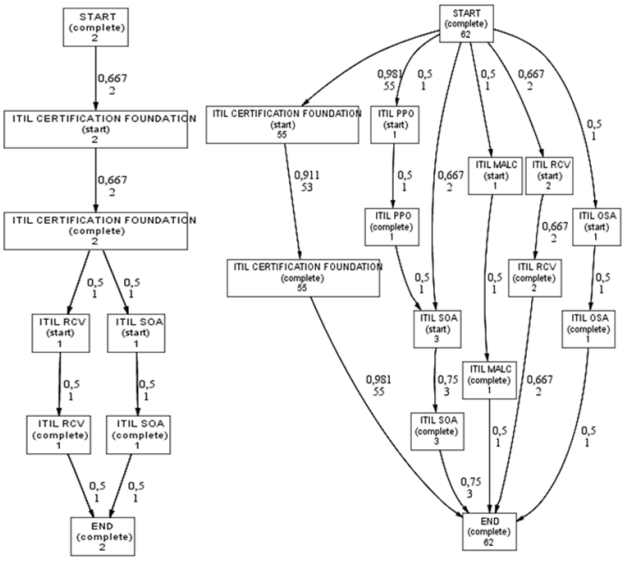
Figure 5. The results returned by the LTL Checker plugin of ProM 5.3 while verifying the Project Management prerequisite

We can see that there are 3420 trainees that satisfy this property and 20 trainees who took the training Project Management–M3 while they didn't take the Project Management -M1 training before.

2) *Conformance Checking*

The *Conformance Checker* plugin supports analysis of the model fitness, precision and structure via log replay, state space analysis, and structural analysis [12]. In what follows, we apply the conformance checker on the training dataset example to verify if the ITIL training program (expressed as a curriculum pattern) is always respected. Our goal is to extract the real ITIL process model as it is followed by trainees during the last three years. We fist apply a filter plugin of ProM on the training log to keep only the traces containing ITIL courses. In a second step, we apply the conformance checker plugin of ProM taking as input the filtered training log and the ITIL training program modelled

as a Petri net. In this program, the training course ITIL Certification Foundation is mandatory. There is also a set of optional training courses: ITIL PPO (Planning, Protection & Optimization), ITIL OSA (Operational Analysis Support) and ITIL MALC (Managing Across the LyfeCycle), ITIL RCV (Release, Control and Validation) and ITIL SOA (Service Offerings & Agreement). From the conforming and non-conforming traces produced by the conformance checking plugin, we mine the process models underlying these two types of traces using the *Heuristic Miner* plugin of ProM (see Figure 6). In Figure 6, the numbers in the boxes indicate the frequencies of the training courses. The decimal numbers along the arcs show the probabilities of transitions between two training courses and the natural numbers present the number of times this order of trainings occur.



(a) Conform training model

(b) Non-conforming training model

Figure 6. ITIL process model mined from the training event log example

The obtained result showed that trainees following a learning path conform to the training ITIL program never took the optional courses ITIL PPO, ITIL OSA and ITIL MALC. Moreover, there are also some trainees who took the same courses as the ITIL program but not in the same order. These results may help training advisors in reviewing the original ITIL training program. They may study the usefulness of some optional courses and propose also new variants of the ITIL training curriculum.

E. Decision Mining in Educational Processes

In a (business) process model, a decision point corresponds to a point where the process is split into alternative paths (e.g., a place with multiple outgoing arcs in a Petri net) [13]. Decision mining (i.e., decision point analysis) aims at the detection of data dependencies that affect the routing of a case. Starting from a process model and a corresponding event log, decision points are identified and data attributes of this log are analysed to determine how case data influence the choices made in the process based on past process executions [13]. In what follows, we analyse choices in training processes to find out which properties of trainee's profile might lead him/her to take certain training paths. To achieve this, we carry out a decision point analysis using the *Decision Miner* plugin of the ProM framework. This later analysis the choice constructs of Petri net process models using the well-known concept of decision trees. The decision miner plugin of ProM takes as an input a Petri net and a corresponding event log. Given the great number of distinct traces in the training log example, we can't use this plugin on the complete training event log. We have first to pre-process this log to reduce its size in order to facilitate the mining of the underlying Petri net model. We started by filtering the training log example, using the *Simple Heuristic Filter* plugin of ProM. We obtained a reduced training log containing employee's training paths referring to the six most frequent training courses in the log. Then, we extract the training process model (represented as a Petri net) corresponding to the filtered training log using the *Alpha Miner* plugin of ProM (see Figure 7).

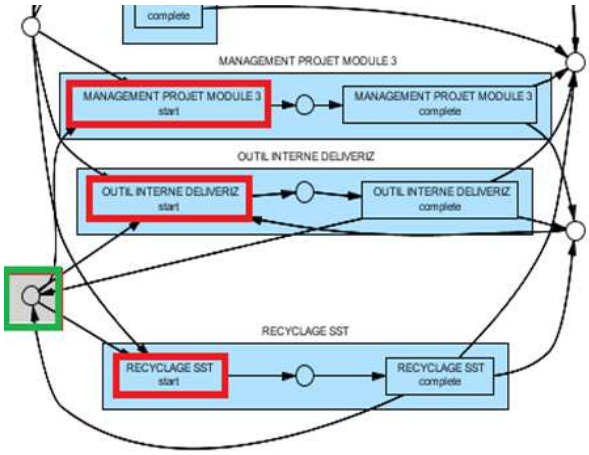


Figure 7. A fragement of the Petri net model underlying the filtered training log containing the six most frequent training courses

The generated Petri net model is then introduced along the filtered training log as inputs in the *Decision Miner* plugin. We choose to study the decision point p_0 (in a green square in Figure 7) which appears after the training course "ITIL certification foundation" to explain the employees' choices between three alternative training paths (Project Management Module 3, Internal Tool Deliviriz or Recycling test). We rely in this analysis on the two case attributes describing an employee profile (i.e., function, number of years of service). Figure 8 shows the decision tree result for the decision point p_0, from which we can now infer the following rule. The training course "Internal Tool

 ISBN: 978-1-61208-415-2

Deliveriz" is chosen by an employee if he/she has been in the company less than 47 months or more than 53 months.

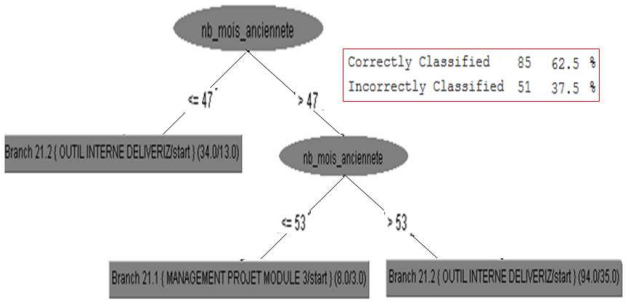


Figure 8. Decision tree result for analysis decision point p0

An employee is more likely to choose the training "Project Management-Module3" if he/she has been in the company between 47 and 53 months. The discovered rule has an accuracy of 62%.

V. CONCLUSION

The aim of our research is to develop an interactive and distributed platform tailored for educational process discovery and analysis. In this paper, we showed how conformance checking, performance analysis and process models enhancement techniques can be used to monitor and improve educational processes in the field of professional trainings. However, performance analysis with Petri net, conformance checking and decision mining, as they are actually implemented in ProM, can't handle heterogeneous and large scale event logs encountered in the professional training field [7, 9]. The adoption of filtering, abstraction or clustering techniques may help reducing the complexity of the discovered process models, and hence the application of advanced analysis techniques [7]. To enhance the usability of our platform, we have also to work on designing an intuitive graphical interface for non-experts that automatically sets parameters and suggests suitable types of analysis.

ACKNOWLEDGMENT

This work is done by Altran Research and Altran Institut in the context of the project PERICLES (http://e-pericles.org/).

REFERENCES

[1] W. M. P. van der Aalst, H. T. de Beer, and B. F. van Dongen, "Process Mining and Verification of Properties: An Approach Based on Temporal Logic," In OTM Conferences, R. Meersman et al., editors, LNCS, 3760 (1):, pp. 130-147, 2005.

[2] W. M. P. van der Aalst et al, "Process mining manifesto," In Business Process Management (BPM) 2011 Workshops Proceedings, pp. 169–194, 2011.

[3] S. Anuwatvisit, A. Tunggkasthan, and W. Premchaiswadi, "Bootleneck mining and petri net simulation in education situations," Conference on ICT and Knowledge Engineering, pp. 244-251, 2012.

[4] A. Bogarín, C. Romero, R. Cerezo and M. Sánchez-Santillán, "Clustering for improving educational process mining,". In Proceedings of the Fourth International Conference on Learning Analytics And Knowledge. ACM, New York, NY, USA, pp. 11-15, 2014.

[5] A. Hicheur Cairns et al., "Towards Custom-Designed Professional Training Contents and Curriculums through Educational Process Mining," IMMM14, pp. 53-58, Jul. 2014, Paris, France.

[6] R. P. Jagadeesh Chandra Bose and W. M. P. van der Aalst, "Process diagnostics using trace alignment: Opportunities, issues, and challenges," Inf. Syst. 37, 2, pp. 117-141, Apr. 2012.

[7] J. Munoz-Gama, J. Carmona, and W. M. P. van der Aalst, "Conformance Checking in the Large: Partitioning and Topology," The 11th International Conference on Business Process Management (BPM 13), pp. 130–145, Aug. 2013, doi:10.1007/978-3-642-40176-3_11.

[8] M. Pechenizkiy, N. Trčka, E. Vasilyeva, W. P. M. van der Aalst, and P. De Bra, "Process Mining Online Assessment Data," The 2nd International Conference on Educational Data Mining (EDM 2009), pp. 279–288, Jul. 2009.

[9] M. Reichert, "Visualizing Large Business Process Models: Challenges, Techniques, Applications," In 1st Int'l Workshop on Theory and Applications of Process Visualization Presented at the BPM 2012, Tallin, pp. 725-736, 2012.

[10] C. Romero, S. Ventura, and E. Garcia, "Data Mining in Course Management Systems: Moodle Case Study and Tutorial," E. Computers & Education, 51(1), pp. 368-384, 2008.

[11] M. Romero and C. Ventura, "Data mining in education," The Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol 3, pp. 12–27, Feb. 2013.

[12] A. Rozinat and W. M. P. van der Aalst, "Conformance checking of processes based on monitoring real behavior," Inf. Syst. 33, 1, pp. 64-95, Mar. 2008.

[13] A. Rozinat and W. M. P. van der Aalst, "Decision mining in prom," In Proceedings of the 4th international conference on Business Process Management (BPM'06), Schahram Dustdar, José Luiz Fiadeiro, and Amit P. Sheth (Eds.), Springer-Verlag, Berlin, Heidelberg, pp. 420-425. 2006.

[14] V. Southavilay, K. Yacef, and R. A. Calvo, "Process mining to support students' collaborative writing," The 3rd International Conference on Educatinal Data Mining (EDM 2010), pp. 257-266, Jun. 2010.

[15] M. Song and W. M. P. van der Aalst, "Supporting Process Mining by Showing Events at a Glance," Seventeenth Annual Workshop on Information Technologies and Systems (WITS'07), In K. Chari, A. Kumar editors, Montreal, Canada, pp. 139–145, Dec. 2007.

[16] N. Trčka and M. Pechenizkiy "From Local Patterns to Global Models: Towards Domain Driven Educational Process Mining," In ISDA 2009, pp. 1114–1119, Dec. 2009, doi:10.1109/ISDA.2009.159.

[17] N. Trčka, M. Pechenizkiy, and W. P. M. van der Aalst, "Process Mining from Educational Data (Chapter 9)," Handbook of Educational Data Mining, CRC Press, pp. 123–142, 2010, doi: 10.1201/b10274-11.

[18] B. van Dongen, H. Verbeek, A. Weijters, and W. P. M. van der Aalst, "The ProM framework: a new era in process mining tool support," The 26th International Conference (ICATPN 2005) LNCS Vol. 3536, pp. 444–454, Jun. 2005, doi:10.1007/11494744_25.

 ISBN: 978-1-61208-415-2

Streamlining the Detection of Accounting Fraud through Web Mining and Interpretable Internal Representations

Duarte Trigueiros
University of Macau, University Institute of Lisbon
Lisbon, Portugal
Email: dmt@iscte.pt

Carolina Sam
Master of European Studies Alumni Association
Macau, China
Email: kasm@customs.gov.mo

***Abstract*—Considerable effort has been devoted to the development of Artificial Intelligence tools able to support the detection of fraudulent accounting reports. Published results are promising but, till the present date, the use of such research has been limited, due to the "black box" character of the developed tools and the cumbersome input task they require. The tool described in this paper solves both problems while improving specificity of diagnostics. It is based on Web Mining and on Multilayer Perceptron classifiers where a modified learning method leads to meaningful representations. Such representations are then input to a features' map where trajectories towards or away from fraud and other features are identified. The final result is a robust Web Mining-based, self-explanatory fraud detection tool.**

Keywords–Type of Information Mining; Knowledge Extraction; Accounting Fraud Mining.

I. INTRODUCTION

Fraud may cost US companies over USD 400 billion annually [1]. Amongst different types of fraud, manipulation of accounting reports is paramount. In spite of measures put in place to detect fraudulent book-keeping, manipulation is still ongoing, probably on a huge scale [1]. Auditors are required to assess the plausibility of manipulated reports. They apply analytical procedures to inspect sets of transactions which are the building blocks of reports. But detecting fraud internally is a difficult task as managers deliberately try to deceive auditors. Most material frauds stem from the top levels of the organization where controls are least effective. The general belief is that analytic procedures alone are rarely effective in detecting fraud [2].

In response to concerns about audit effectiveness in detecting fraud, quantitative techniques are being applied to the modelling of relationships underlying published reports' data with a view to discriminate between fraudulent and non-fraudulent reports [3][4]. Such external, *ex-post* approach would be valuable as a tool in the hands of users of published reports such as investors, analysts and banks. Artificial Intelligence (AI) techniques are likewise being developed to the same end. Detailed review articles covering this research are available [5][6].

A discouraging fact is that analysts do not use AI tools designed to help detecting accounting manipulation. This is largely due to the fact that such tools are "black boxes" where results cannot be explained using the viewpoint, language and expertize of analysts [2]. Since analysts are responsible for their decisions, tools they may use to support decisions must be transparent and self-explanatory. Moreover, extract, transform and load (ETL) tasks required by such AI tools are time-consuming and difficult to automate in this case. The paper describes work-in-progress seeking to overcome the above limitations. Web Mining is first employed to find, download and store accounting data. Then, fraud and two other attributes known to widen fraud propensity space are predicted by three Multilayer Perceptron (MLP) classifiers where a modified learning method leads to internal representations similar to financial ratios, readily interpretable by analysts. Such ratios then input a features' map where trajectories towards or away from fraud and other features are visualized. Diagnostic interpretation is further enhanced with the display of cases similar to those being analyzed.

The objective of the tool is not so much to innovate but to streamline a well-known but opaque and cumbersome practice. Its sole original contribution is the strict adherence to users requirements including a new MLP training method leading to transparent diagnostic. Fraud detection covers many types of deception: plagiarism, credit card fraud, medical prescription fraud, false insurance claim, insider trading, accounting reports' manipulation and other [12][13]. Frameworks used in the detection of, say, credit card fraud (such as Game Theory), are not necessarily efficient in detecting other types of deception. Neural Networks are widely used in research devoted to the detection of accounting fraud [7][8][9][10][11] and reported performance is satisfactory.

Section II describes data and models while offering extensive methodological details. Section III reports preliminary results and presents the architecture of the tool to be deployed. Section IV discusses expected benefits.

II. METHODOLOGY

A. Accounting Information

An accounting report is a collection of monetary amounts with an attached meaning: revenues of the period, different types of expenses, asset values at the end of the period, liabilities and others. Companies' reports are obtained via a process involving recognition, adjustments and aggregation into "accounts", of all meaningful transactions

 ISBN: 978-1-61208-415-2

occurring during a given period. The resulting set of reports is made available to the public together with notes and auxiliary information.

Accounting reports are extremely efficient in revealing financial position. It is possible, for instance, to accurately predict bankruptcy more than one year before the event [14]. The direction of future earnings (up or down) is also predictable [15]. Such efficiency in conveying useful information is the ultimate reason why accounts are so often manipulated by managers.

Financial analysis of a company is typically based on the comparison of two monetary amounts (hereafter referred to as "items") taken from published reports. For instance, when a company's net income at the end of a given period is compared with assets required to generate such income, an indication of "Profitability" emerges. Pairs of items are often expressed in the form of a single value, their ratio. Since the dimension effect is similar for all items taken from the same company and period, dimension cancels out when a ratio is formed. Thus, ratios compare features such as performance of companies of different dimension [16]. Ratios are also used to detect fraud [3][4]. Indeed, most analytical tasks involving accounting information require the use of appropriately chosen ratios so that companies of different sizes can be compared while their financial features are highlighted. In this paper, an MLP training method is described whereby ratios with optimal performance characteristics are uncovered.

B. Web Mining of XBRL-encoded reports

Until recently, accounting reports were published in a variety of formats including PDF, MS Word and MS Excel. This forced users and their supporting tools into a significant amount of interpretation and manual manipulation of meta-data and led to inefficiencies and costs. From 2010 on, the Securities and Exchange Commission (SEC) of the US, as well as United Kingdom's Revenue & Customs (HMRC) and other regulatory bodies, require companies to make their financial statements public using the XML-based eXtensible Business Reporting Language (XBRL). Users of XBRL now include securities' regulators, banking regulators, business registrars, tax-filing agencies, national statistical agencies plus, of course, investors and financial analysts worldwide [17]. XML syntax and related standards such as XML Schema, XLink, XPath and Namespaces are all incorporated into XBRL, which can thus extract financial data unambiguously. Communications are defined by metadata set out in taxonomies describing definitions of reported monetary values as well as relationships between them. XBRL thus brings semantic meaning into financial reporting, promoting harmonization, interoperability and greatly facilitating ETL tasks. Web Mining of financial data is now at hand.

The initial module of the tool proposed here carries out Web Mining of XBRL content. The user first defines a selection criteria namely an industrial group, a range of assets' dimensions or simply a set of companies' codes. Then specific Web locations are searched. In the US, for instance, one such location is the SEC repository (known as EDGAR) of "fillings" of companies' reports and other data. Reports obeying stipulated criteria are downloaded and items required by the analysis are stored.

C. Data and Models

After mining and storage, three MLP are set to separately predict fraud vs no-fraud cases plus two other attributes known to widen fraud propensity space. Inputs to each of the three MLP are collections of items which were utilized as numerators or denominators of ratios in published research, namely:

- fraudulent vs non-fraudulent reports [3][4]
- bankrupted vs solvent companies [14]
- profits-up vs profits-down one year ahead [15].

Collections of items are taken from the same company reports (instance j) and may include 8 to 12 items. Both the actual period, t, and previous period, $t-1$, are collected. Items which assume positive and negative values such as net income are replaced by their two positive-only components. Input variables and target attributes used in the training and testing of the three MLP are extracted from three sources:

- "Compustat", the *de facto* repository of US companies' financial information, made available by Standard & Poor's;
- a total of 1,300 Accounting and Auditing Enforcement Releases (AAER) issued by the SEC, identifying a given set of accounts as fraudulent [4], covering the period 1983-2013, are made available by the Haas School of Business (Centre for Financial Reporting and Management) at the University of California, Berkeley;
- a list of 750 US bankruptcies covering the period 1992-2005, is made available by Professor Edward Altman from New York University.

Before training, MLP architectures consist of up to 12 inputs corresponding to collections of items just mentioned, one hidden layer with 6 nodes and two symmetrical output nodes. Hyperbolic tangents (threshold functions symmetrical around zero) are used as transfer functions in all nodes. During training, balanced matching of cases is carried out using same industry, same size (Total Assets decile) and same year companies with opposite class attribute. Training- and testing-sets are equally matched. Financial companies such as banks are excluded.

D. Knowledge Extraction

Studies on the statistical characteristics of items from accounting reports brought to light two facts. First, in cross-section the probability density function governing such items is nearly lognormal. Second, items taken from the same set of accounts share most of their variability as the dimension effect is prevalent [16]. Thus, the variability of logarithm of item i from set of accounts j, $\log x_{ij}$, is explained as the dimension effect σ_j, which is present in all items from j, plus some residual variability ε_i particular to item i:

$$\log x_{ij} = \mu_i + \sigma_j + \varepsilon_i \quad (1)$$

 ISBN: 978-1-61208-415-2

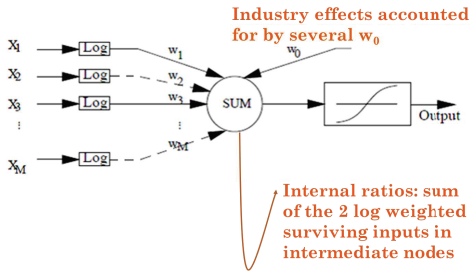


Figure 1. Ratio x_{kj}/x_{ij} of items k and i from report j is formed in MLP hidden node as a log representation $\log x_{kj} - \log x_{ij}$ because synaptic weights assume symmetrical values: $w_k = -w_i$.

μ_i is the item- and industry-specific expected value. It is thus clear why ratios formed with two items from the same set of accounts are effective in conveying financial information: the dimension effect, σ_j, cancels out when a ratio is formed. Median ratios are industry expectations while deviations from expectation observed in company j reveal how well j is doing no matter its dimension. For instance, in a given industry the median ratio of net income to assets is, say, 0.15. Any company with a ratio above 0.15, no matter small or large, is doing better than the industry.

When analyzing features such as Solvency or the likelihood of fraud, financial analysts need to know which ratios are at work, their position in relation to expectations and in which direction they are moving. In order to respond to the first of such demands, MLP training includes the competitive pruning of synaptic weights linking inputs, the $\log x_i$ in (1), to hidden nodes so that, at the end of training, only the two most relevant weights in each hidden node are permitted to survive. In a later phase, nodes also compete for survival. MLP training encompasses 5 steps:

Step	I	No penalization of synaptic weights.
	II	All hidden-node synaptic weights are equally penalized.
	III	Penalization of less relevant weights but two, one node at a time.
	IV	Zero-valued weights, all but two in each node, are pruned.
	V	Node-pruning.

In this way, internal representations similar to ratios in log space are formed inside each surviving hidden node. Here, the term "internal representation" refers to values assumed by each hidden node after summation but before transfer function, as depicted in Figure 1. The fact that each node succeeds in forming a ratio is visible through the examination of its two surviving synaptic weights: they are of similar magnitude but with opposite sign so that, after summation, a log-ratio (a difference between two $\log x_j$) is formed. Although absolute values of the two surviving weights in each hidden node are not much different from one another, they differ across nodes. Such difference reflects the importance of each node for the final classification performance.

Internal representations tend to assume the form of ratios because instances used in MLP training greatly differ in dimension while the attribute to be predicted is indeed predictable. Hidden nodes thus tend to self-organize themselves into dimension-independent variables, efficient in predicting such attribute. And since only two of the synaptic weights in each node, the most explanatory of them, are allowed to survive, weights' final values tend to assume symmetrical values so that their summation is indeed dimension-free. Representations thus mimic ratios and can be interpreted similarly.

After appropriate ratios are selected, analysts interpret their observed, company-specific deviations from industry expectation. Correspondingly, each hidden node in the MLP has a set of dummy inputs assuming the value of 1 or 0 depending on the industrial group of instance j. In this way, expected μ_i from (1) are also modelled and accounted for inside each hidden node. Since node outputs and attributes' classes are both balanced, the effect of industry dummies is to subtract industry-specific log-ratio standards from internal representations thus making them similar to a difference of two ε_i in (1). Such difference is, in log space, what analysts seek when they compare a ratio with its industry expectation.

E. Trajectories in a Features' Map

Finally, analysts observe in which direction ratios move. Internal representations are likewise input to a 2-dimensional Kohonen Features Map with 8×8 nodes. MLP outputs (transformed to become 0-1 variables) are combined with Prevalence numbers (prior probabilities of fraud) so as to approach posterior probabilities of fraud given observed features. After training, clusters are formed in the Kohonen Map, denoting identifiable features such as Solvency, Profitability, Fraud or their opposites. Visual examination of features' maps facilitates interpretation, both proximity to a given cluster and trajectories towards or away from clusters being informative.

F. Outputs to be Used by Analysts

When analysing a company's reports, analysts base their diagnostic on several concurring pieces of evidence, in favour or against *a priori* hypotheses. On the other hand, extant research on accounting manipulation suggests that fraudulent numbers lead to detectable imbalances in financial features. For instance, income may increase without the corresponding, usual increase in free cash. The selection of the two attributes complementing fraud (bankruptcy and profit direction) responds to imbalances mentioned in published research [3][4] and to the need, in the part of analysts, to examine concurring facts. Each company being analysed generates two sets of results corresponding to time periods $t-1$ and t. Output to analysts consists of the following:

1) Three posterior probabilities: fraud, default and profits going down, with a sign indicating the direction of their change from $t-1$ to t.
2) The 9 most significant values internal representations assume at period t, three from each MLP, expressed as percent increase or decrease in relation

 ISBN: 978-1-61208-415-2

to industry expectations, with a sign indicating the direction of change from $t-1$ to t.

3) Visualization of features and their trajectories from $t-1$ to t, allowing the detection of trends towards a given cluster.
4) Identification of companies neighbouring, in the features map, the company being analysed.

III. Preliminary Results, Deployment

MLP test-set performance is similar to that reported for other AI tools: 80% success in detecting fraud (6 surviving nodes), 96% success in detecting bankruptcy (5 nodes) and 78% success in predicting earnings' increase one year ahead (6 nodes). Errors are balanced: Type II error (the most expensive in this case) is reduced in relation to published research while Type I error is increased. The number of variables and synaptic weights engaged in modelling is less than half of that reported in the literature. Robustness is expected to be higher. In the downside, ratios that are formed and MLP performance both depend on broad industry type.

The tool has been set up using a variety of packages and languages; it is to be deployed as a Java-based set of modules as depicted in Figure 2. With the exception of the MLP algorithm, the analytical core will be written in R-language.

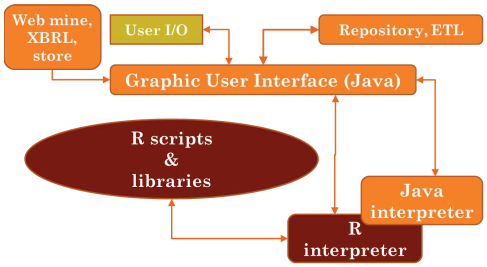


Figure 2. Architecture of the tool to be deployed.

IV. Conclusion and Future Work

Till the present date, the use of AI tools to help in the detection of manipulated accounts has been limited due to difficulties in extraction and put in place of data and also due to the "black box" nature of such tools. The present work-in-progress aims at solving both problems, producing automated and interpretable diagnostics. In the hands of analysts, the tool is self-explanatory, not just pointing out companies likely to have committed fraud but showing, rather than hiding, reasons behind such diagnostic.

The tool illustrates a case of close alignment between users' needs and algorithmic characteristics. The tool is also an example of Knowledge Extraction whereby explanatory variables are discovered amongst many candidates so that a discriminating task is carried out with optimal performance. The choice of the algorithm, the MLP, was dictated solely by its ability to form internal representations. Neither an increased performance nor the testing of novel AI techniques is the goal here. The goal is to build a usable tool, an apparently simple task but which, in this particular subject area, has eluded research effort during the last 20 years. Thus, the ultimate test is yet to be carried out, namely whether analysts will use the tool or not.

Acknowledgment

Research sponsored by the Foundation for the Development of Science and Technology of Macau, China.

References

[1] M. Nigrini, Forensic Analytics: Methods and Techniques for Forensic Accounting. John Wiley and Sons, 2011.

[2] W. Albrecht, A. C., and M. Zimbelman, Fraud Examination. Mason, OH: South-Western Cengage Learning, 2009.

[3] M. Beneish, "The Detection of Earnings Manipulation," Financial Analysts Journal, vol. 55, no. 5, 1999, pp. 24–36.

[4] P. Dechow, W. GE, C. LARSON, and R. Sloan, "Predicting Material Accounting Misstatements," Contemporary Accounting Research, vol. 28, no. 1, 2011, pp. 17–82.

[5] E. Ngai, Y. Hu, Y. Wong, Y. Chen, and X. Sun, "The Application of Data Mining Techniques in Financial Fraud Detection: a Classification Framework and an Academic Review of Literature," Decision Support Systems, vol. 50, no. 3, 2011, pp. 559 – 569.

[6] A. Sharma and P. Panigrahi, "A Review of Financial Accounting Fraud Detection Based on Data Mining Techniques," International Journal of Computer Applications, vol. 39, no. 1, 2012.

[7] E. Kirkos, S. Charalambos, and Y. Manolopoulos, "Data Mining Techniques for the Detection of Fraudulent Financial Statements," Expert Systems with Applications, vol. 32, 2007, p. 995–1003.

[8] W. Zhou and G. Kapoor, "Detecting Evolutionary Financial Statement Fraud," Decision Support Systems, vol. 50, 2011, pp. 570–575.

[9] P. Ravisankar, V. Ravi, G. Raghava Rao, and I. Bose, "Detection of Financial Statement Fraud and Feature Selection Using Data Mining Techniques," Decision Support Systems, vol. 50, no. 2, 2011, pp. 491–500.

[10] F. H. Glancy and S. B. Yadav, "A Computational Model for Financial Reporting Fraud Detection," Decision Support Systems, vol. 50, no. 3, 2011, pp. 595–601.

[11] S.-Y. Huang, R.-H. Tsaih, and F. Yu, "Topological Pattern Discovery and Feature Extraction for Fraudulent Financial Reporting," Expert Systems with Applications, vol. 41, no. 9, 2014, pp. 4360 – 4372.

[12] L. V. S.-M. K. Phua, C. and R. Gayler, "A Comprehensive Survey of Data Mining-based Fraud Detection Research," 2005, Clayton School of Information Technology, Monash University.

[13] U. Flegel, J. Vayssire, and G. Bitz, "A State of the Art Survey of Fraud Detection Technology," in Insider Threats in Cyber Security, ser. Advances in Information Security, C. W. Probst, J. Hunker, D. Gollmann, and M. Bishop, Eds. Springer US, 2010, vol. 49, pp. 73–84.

[14] E. Altman, "Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy," The Journal of Finance, vol. 23, no. 4, Sep. 1968, pp. 589–609.

[15] J. Ou and S. Penman, "Financial Statement Analysis and the Prediction of Stock Returns," Journal of Accounting and Economics, vol. 11, no. 4, 1989, pp. 295–329.

[16] S. McLeay and D. Trigueiros, "Proportionate Growth and the Theoretical Foundations of Financial Ratios," Abacus, vol. XXXVIII, no. 3, 2002, pp. 297–316.

[17] T. Dunne, C. Helliar, A. Lymer, and R. Mousa, "Stakeholder Engagement in Internet Financial Reporting: The Diffusion of {XBRL} in the {UK}," The British Accounting Review, vol. 45, no. 3, 2013, pp. 167 – 182.

 ISBN: 978-1-61208-415-2

Exponential Moving Maximum Filter for Predictive Analytics in Network Reporting

Bin Yu, Les Smith, Mark Threefoot
Advanced Technology, CTO Office
Infoblox Inc.
Santa Clara, California, USA
e-mail: {biny,lsmith,mthreefoot}@infoblox.com

***Abstract*—In networking industry, there are various services that are mission critical. For example, DNS and DHCP are essential and are common network services for a variety of organizations. An appliance that provides these services comes with a reporting system to provide visual information about the system status, resource usage, performance metrics, and trends, etc. Furthermore, it is desirable and important to provide prediction against these metrics so that users can be well prepared for what is going to happen and prevent downtime. Among the predictive measures, there are multiple metrics to reflect peak or maximum values such as peak volume or resource usage in networking. The peak value prediction is critical for the IT managers to ensure its organization is ahead of the cycles in terms of the network capacity and disaster recovery. There have been many algorithms and methods for prediction of trended time series data. However, peak values often do not fall into a trend by nature. The traditional trend prediction methods do not perform well against this type of data. In this paper, we present a novel filtering algorithm named "Exponential Moving Maximum" (EMM), this filter is used before applying a prediction algorithm against peak time series data. We also provide some experimental results on real data as a comparison to show that the prediction method has better accuracy when EMM filtering is applied to certain categories of networking data.**

Keywords-predictive analytics; trend forecasting; networking reporting; time series data; sequential pattern mining

I. INTRODUCTION

There have been a number of methods that can do trend prediction or forecasting on time series data by removing so called non-stationarity or noise. Simple moving average is the most basic technique that averages the last n observations of a time series [1][2]. It is appropriate only for very short or irregular data sets, where features like trend and seasonality cannot be meaningfully determined, and where the mean changes slowly. Exponential smoothing, such as the Holt-Winters method, is a more complex moving average method that involves parameters reflecting the level, trend and seasonality of historical data. It usually gives more weight to recent data [2]-[6]. An even more complex class of moving average models, autoregressive moving average (ARMA) [2][6]-[8] is capable of reflecting autocorrelations inherent in data. It can out-perform exponential smoothing when the historical data period is long and data is nonvolatile. But it doesn't perform as well when the data is statistically messy. The typical application of this forecasting technique is in marketing for which J. Armstrong *et al.* had a review on many methods in their publication [9].

One of the most active research areas employing trend prediction is stock market forecasting. Therefore, many researchers have applied different analysis methods to do stock trend prediction, including associative rule based approaches, chart pattern recognition, template matching, neural networks and SVM [10][11]. K. Wu *et al.* recently presented a method to predict stock trend with k-means clustering algorithms [12] in identifying patterns within a sliding window. However the complexity of the algorithm poses a limitation for methods that are used in real time applications.

Time series data generated by a network service system such as DNS and DHCP servers often contains useful non-stationarity of which an example is illustrated in Figure 1 that is a time series DNS query data with hourly maximum or peak values for a period of 270 days. Users, typically from IT departments, are interested in seeing the trend of peak value data and, furthermore, to know the prediction for near future. Therefore, they can have means to assess the capacities of their network allowing purchase and deployment of new equipment to meet expected demand with out having to over provision.. When a traditional prediction algorithm is used with this data, the information about the local maximums will unfortunately get lost.

In Section II, we present the algorithm of exponential moving maximum and its memory complexity. Section III provides the command lines and workflow for the integration with Splunk [13]. We present the experimental results with comparison in Section IV. The conclusion is presented in Section V.

II. EXPONENTIAL MOVING MAXIMUM FILTER

The EMM filter is used to aggregate historical values with a maximum aggregator so that the effect of these values can be taken into account by subsequent values whilst applying a magnitude decaying exponential along with time. That's similar to one of the special cases in ARIMA that's exponential moving average [14]-[16] where historical values are aggregated into the following values but with an average aggregator. The EMM filter can be defined as

 ISBN: 978-1-61208-415-2

$$y_k = \max_{0 \le i \le k} \{\alpha^{\frac{i}{w}} x_{k-i}\}$$

where

$$\alpha \in [0,\ 1.0]$$

is an inheritance parameter and w is a filtering window size. In the case when

$$\alpha = 0, y_k = x_k$$

and when

$$\alpha = 1, y_k = \max_{0 \le i \le k} \{x_i\}$$

If

$$y_k = \alpha^{\frac{m}{w}} x_{k-m}$$

then x_{k-m} is called the bubble point of y_k and m is the bubble distance. Figure 2 illustrates the relationship of the parameters in which the original value x will have a contribution on the magnitude of αx for the filtered value at a future position that is w distance away from x. It shows the future impact of a value is decayed exponentially over time.

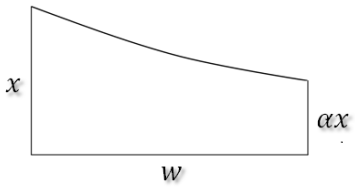


Figure 2. EMM filter parameters.

Figure 3 shows an example of EMM filtering over the hourly peak DNS query time series data. The peak values that are local maximums can become bubble points that over shadow the following non local maximal data points. The red curves show the EMM filtering output which effectively preserves the historical information of peak values and can contribute to the prediction of future peak values.

It can be proven that

$$\begin{aligned} y_k &= \max_{0 \le i \le k} \{\alpha^{\frac{i}{w}} x_{k-i}\} \\ &= \max\left(x_k, \alpha^{\frac{1}{w}} x_{k-1}, \alpha^{\frac{2}{w}} x_{k-2}, \dots, \alpha^{\frac{k}{w}} x_0\right) \\ &= \max\left[x_k, \alpha^{\frac{1}{w}}\left(x_{k-1}, \alpha^{\frac{1}{w}} x_{k-2}, \dots, \alpha^{\frac{k-1}{w}} x_0\right)\right] \\ &= \max\left(x_k, \alpha^{\frac{1}{w}} \max_{0 \le i \le k-1} \{\alpha^{\frac{i}{w}} x_{k-1-i}\}\right) \\ &= \max\left(x_k,\ \alpha^{\frac{1}{w}} y_{k-1}\right) \end{aligned}$$

This is the EMM representation in a recursive format that simplifies memory complexity to $O(1)$ for implementation.

III. SPLUNK CUSTOMIZATION

Splunk is a commercial software solution that provides archiving, indexing and analytics functions to machine generated data such as system logs and network data. One of its analytical functions is called *predict()* that can do forecasting based on a series of historical data points. As a platform, Splunk provides an SDK for users to develop custom commands as plug-ins. A custom command named *emm* is developed in Python and plugged into the Splunk system. The syntax of the command is as follows.

```
emm <variable_to_predict> [inheritance=i] [window=w]
```

where i is a floating point value between 0 and 1.0 to represent α and w is an integer value equal to the timespan. For instance, for hourly time series data, w=720 has a window size of one month.

In addition, the native Splunk command *predict()* is customized into a new command *forecast()* with following syntax.

```
forecast <variable_to_predict> [AS <newfield_name>]
[<forecast_option>]
```

The forecast_option is similar to the options for the command *predict()* except for some fields that have different default values customized.

With the custom commands deployed into Splunk app, the prediction process with EMM filtering can be pipelined similar to most of the Splunk queries. A sequence diagram is given in Figure 4 with the steps listed as follows.

1. Load event data files into Splunk system.
2. Fire a query to start the prediction process.
3. The query starts from event aggregation with use of Splunk aggregation functions.
4. Apply EMM filtering on aggregated time series data.
5. Further aggregate EMM output into a coarse level desired by prediction objectives.
6. Execute custom forecast command to get prediction result.
7. Visualize prediction result with Splunk visualization functions.

A sample query command pipe in Splunk is given as follows.

```
source="dns.txt" | rex
"^(?P<date>[^\\t]+)\\t(?P<dns>.+)" | timechart
span=10m max(dns) as dns | emm dns inheritance=0.7
window=4320| timechart span=mon max(emm) as emm
| forecast emm future_timespan=3
```

The output is shown in Figure 5 in which the top section is for command input and the lower part shows the EMM filtering and prediction results.

IV. EXPERIMENT AND COMPARISON

Infoblox is a company that provides DNS, DHCP and IP address management (DDI) appliances for network automation [17]. Its Trinzic DDI™ series is distributed with a Grid Master™ and a number of Grid Members™. The

reporting appliance can be one of the members that collects, archives and analyzes the network data across many members. About 18 month data is collected from two separate customers who are using Infoblox's Trinzic™ Reporting appliance. The data includes number of DNS queries aggregated every 10 minutes, number of DHCP leases aggregated every one minute, DNS server cache hit rate, and system CPU usage history. For each category, the data is segmented into a range of 12 months with a window sliding by month. To simplify the computation, EMM filter is applied on hourly maximum and the prediction is executed on monthly maximum data points. The experiment will use the first consecutive 9 month data from each segment to predict the values for the next three months. The prediction results will then be compared to the reserved three month data for accuracy calculation. The prediction error is defined as a mean squared error

$$MSE = \frac{1}{m}\sum_{k=1}^{m}\frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

where $\hat{Y}_i$ is the prediction value of Y_i at the $(i+9)$th month and $i = 1, 2, 3$. m is the number of sliding steps. Based on MSE, the comparison metric is defined as

$$C = \frac{MSE_{EMM}}{MSE}$$

where MSE_{EMM} is the prediction error with use of EMM filter and MSE is the prediction error without use of EMM. It is apparent that $C < 1$ means accuracy improvement.

First of all, use the Splunk built-in prediction function to analyze the data and provide prediction results in the protocol set above. The raw data is in system log format that is loaded and parsed by a custom regular expression to extract the values from raw events. The software does aggregation on event data to generate time series sequences before applying its *predict()* command. The prediction results can be visualized by its built-in GUI or exported into a text file and visualized separately as illustrated in Figure 6, where the prediction results are shown in the gray area. Secondly, we apply the EMM filtering before invoking the *predict()* command. The EMM filtering results are superimposed on to the raw DNS data that is shown in Figure 6 and highlighted in red in Figure 7. Its prediction results are illustrated in the gray area in Figure 7. Unlike the version without use of EMM in Figure 6 that shows a relatively flat trend, the version with use of EMM in Figure 7 effectively shows the upward trend that matches real data. The same experiments are conducted on all of four categories of network data. For comparison, the above defined C values are calculated and listed in Table 1. The C values for DNS query and DHCP lease data are much smaller than 1 which proves a significant improvement in prediction accuracy. On the other side, the prediction accuracy improvement on the DNS hit rate data is not significant and the accuracy decreases on the CPU data. We will try to provide some explanation in next section.

TABLE I. ACCURACY COMPARISON

Test Data	Comparison Metric C
DNS Query	0.07
DHCP Lease	0.33
DNS Hit Rate	0.98
CPU	1.21

V. CONCLUSION

Many prediction algorithms and methods have been experimented against the time series data that contains non-stationary peak values with poor performance. A preprocessing approach is proposed in this paper as well as an exponential moving maximum filter that can preserve local maximum values from the historical data and make prediction more accurate, meaningful and useful. An example EMM plug-in has been tested with use of Splunk software that provides an ease-of-use user interface and SDK for customization with plug-ins. The same method and algorithm can be used together with other prediction tools or software. The experimental results show that using the EMM filter provides better prediction accuracy on DNS and DHCP data compared to a traditional prediction algorithm provided by some commercial software. Unlike the experiments for DNS and DHCP volume data, the experiments on cache hit rate data and CPU data either show no improvement or present slightly worse accuracy. The possible explanation based on the sample data is that the spikes in cache hit and CPU data look more like real noise than trended peaks in DNS and DHCP volumes. This concludes that EMM should only be applied to the time series data that contains non-stationality which is intrinsically not random noise. Further experiments are needed to add EMM filter into other traditional predication algorithms and methods.

REFERENCE

[1] E. Booth, J. Mount, and J. Viers: "Hdrologic Variablility of the Cosumnes River Floodplain," San Francisco Estuary and Watershed Science, vol. 4(2), 2006.

[2] S. Makridakis, S. Wheelwright, and R. Hyndman, "Forecasting: Methods and Applications (3rd ed.)," New York: John Wiley & Sons, 1998.

[3] J. Armstrong, Principles of Forecasting: A Handbook for Researchers and Practitioners (Section 8: "Extrapolation of time-series and cross-sectional data"). Boston, MA: Kluwer Academic, 2001.

[4] E. Gardner, "Exponential Smoothing: the State of the Art," Journal of Forecasting, vol. 4, 1985, pp. 1-28.

[5] E. Gardner, "Exponential smoothing: The state of the art – Part II." International Journal of Forecasting, vol. 22, 2006, pp. 637-677.

 ISBN: 978-1-61208-415-2

[6] D. Montgomery, C. Jennings, and M. Kulahci, Introduction to Time Series Analysis and Forecasting. Hoboken, N.J.: 34.: Wiley-Interscience, 2008.
[7] G. Box, G. Jenkins, and G. Reinsel, "Time Series Analysis: Forecasting and Control," 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 1994.
[8] S. Makridakis, M. Hibon, "ARMA Models and the Box–Jenkins Methodology". Applied Econometrics (Second ed.). Palgrave MacMillan, ISBN 978-0-230-27182-1, 2011, pp. 265–286.
[9] J. Armstrong, R. Brodie, and S. McIntyre, Forecasting Methods for Marketing: Review of Empirical Research, International Journal of Forecasting, Vol. 3, 1987, pp. 355-376.
[10] C. Slamka, B. Skiera, and M. Spann, "Prediction market performance and market liquidity: A comparison of automated market makers," IEEE Transactions on Engineering Management, vol. 60, 2013, pp. 169-185.
[11] A. Zhu and X. Yi, "The comparisons of four methods for financial forecast," in Proceedings of IEEE International Conference on Automation and Logistics, 2012, pp. 45-50.
[12] K. Wu , Y. Wu and H. Lee, "Stock Trend Prediction by Using K-Means and AprioriAll Algorithm for Sequential Chart Pattern Mining," Journal of Information Science and Engineering, vol. 30, 2014, pp. 653-667.
[13] http://www.splunk.com, 2015.
[14] R. Brown, Exponential Smoothing for Predicting Demand, Cambridge, Massachusetts: Arthur D. Little Inc. 1956, pp. 15.
[15] J. Lucas and M. Saccucci, "Exponentially Weighted Moving Average Control Schemes: Properties and Enhancements," Technometrics, vol. 32, 1990, pp. 1-29.
[16] C. Lowry, W. Woodall, C. Champ, and S. Rigdon, "A Multivariate Exponentially Weighted Moving Average Chart," Technometrics, vol. 34, 1992, pp. 46-53.
[17] http://www.infoblox.com, 2015.

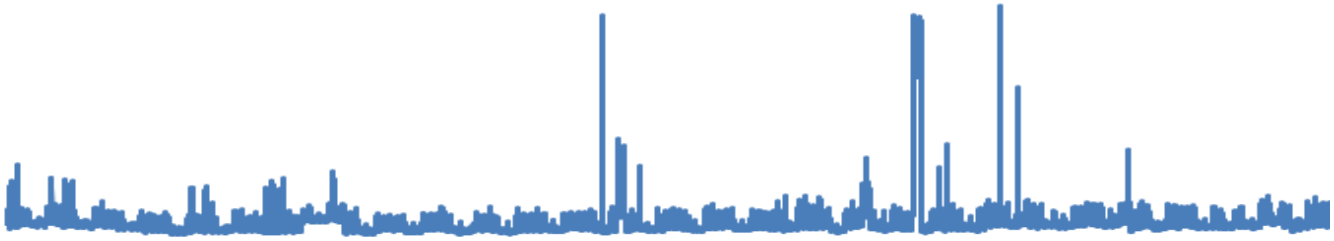


Figure 1. Customer DNS volume data example with meaningful non-stationality.

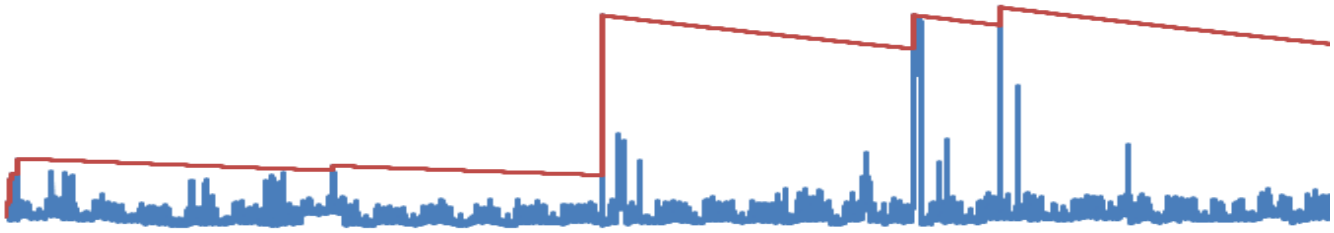


Figure 3. EMM filtering example.

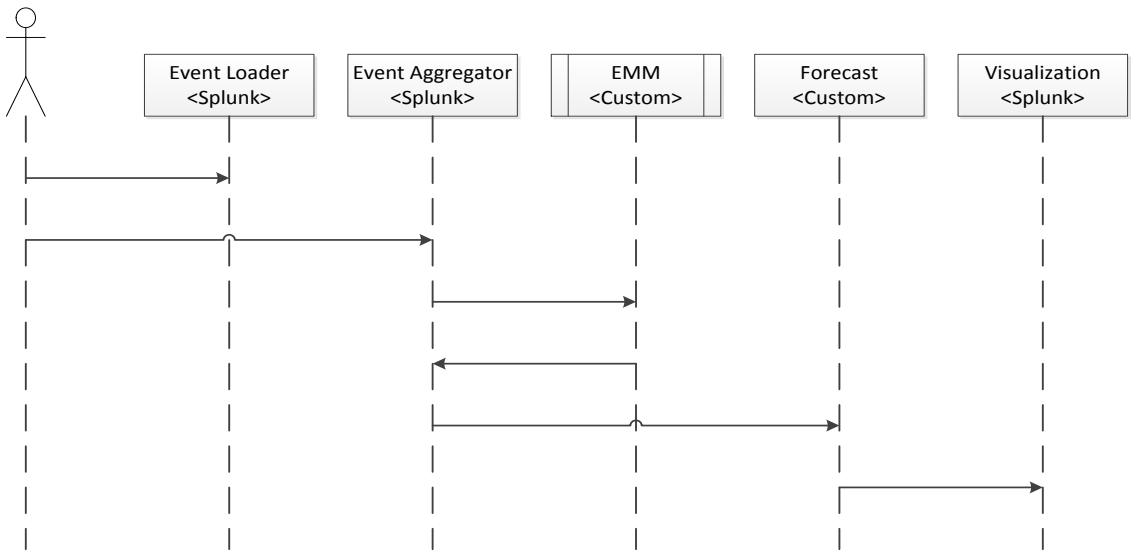


Figure 4. Sequence diagram of prediction process with EMM on Splunk.

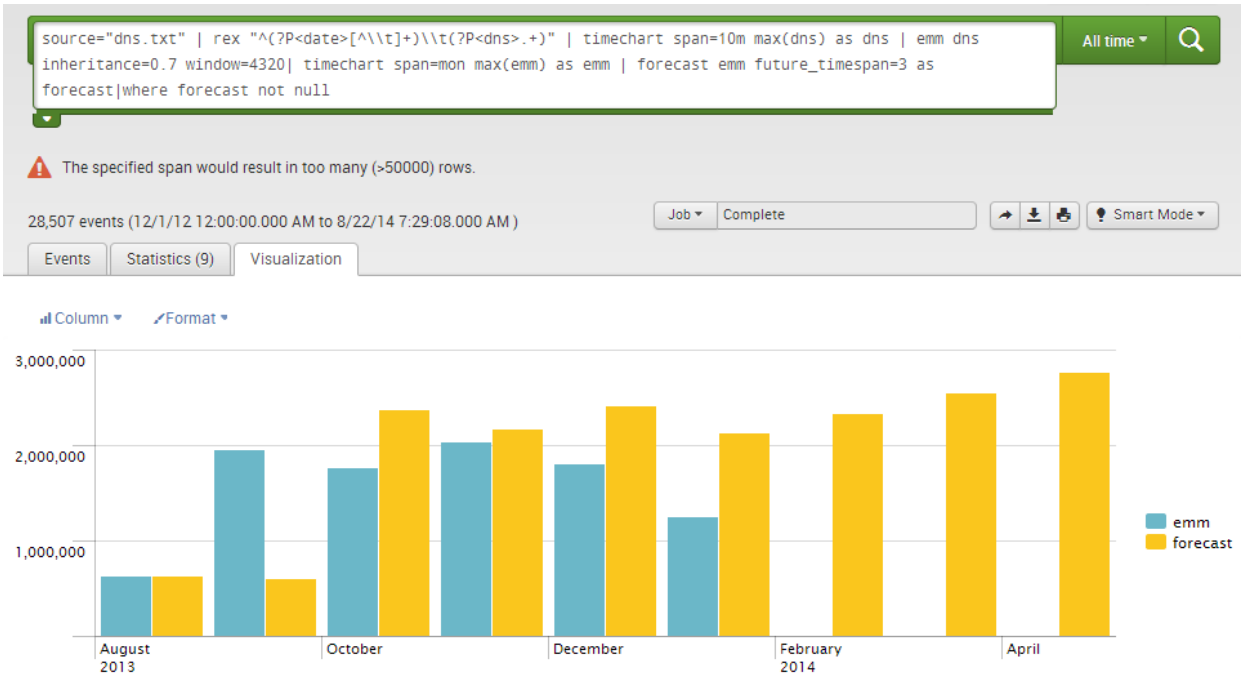


Figure 5. A screen snapshot of the web GUI of Splunk prediction with EMM plug-in.

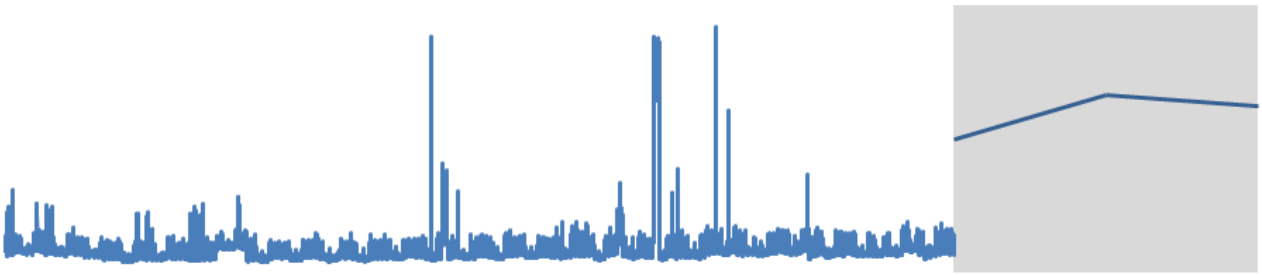


Figure 6. Prediction without use of EMM filter.

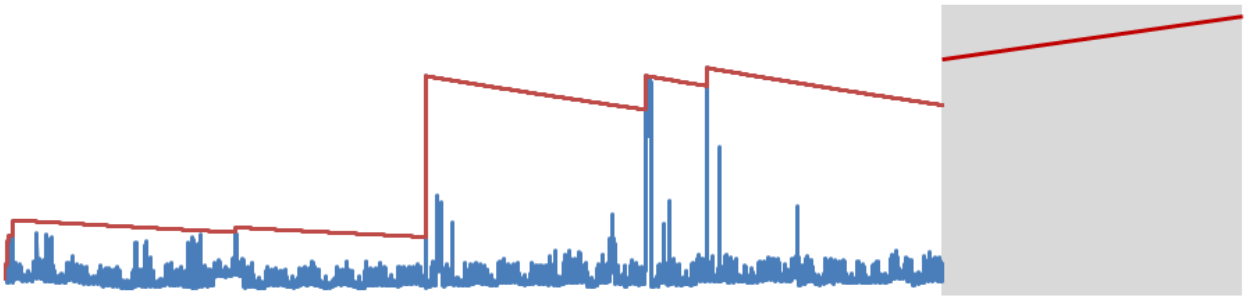


Figure 7. Prediction with use of EMM filter.

 ISBN: 978-1-61208-415-2

Automatic KDD Data Preparation Using Multi-criteria Features

Youssef Hmamouche*, Christian Ernst† and Alain Casali*
*LIF - CNRS UMR 6166, Aix Marseille Université, Marseille, France
Email: firstname.lastname@lif.univ-mrs.fr
†CMP - SGC, Ecole des Mines de St Etienne and LIMOS, CNRS UMR 6158, Gardanne, France
Email: ernst@emse.fr

***Abstract*—We present a new approach for automatic data preparation, applicable in most Knowledge Discovery and Data Mining systems, and using statistical features of the studied database. First, we detect outliers using an approach based on whether data distribution is normal or not. We outline further that, when trying to find the most appropriate discretization method, what is important is not the law followed by a column, but the shape of its density function. That is why we propose an automatic choice for finding the best discretization method based on a multi-criteria (Entropy, Variance, Stability) analysis. Experimental evaluations validate our approach: The very same discretization method is never always the most appropriate.**

Keywords–Data Mining; Data Preparation; Outliers; Discretization Methods.

I. Introduction and Motivation

Data preparation can be performed according to different method(ologie)s [1]. However, this task has not been developed greatly in the literature: The single mining step is more often emphasized. Moreover, it focuses most of the times on a single parameter: discretization method [2], outlier detection [3], null values management, *etc.*. Associated proposals only highlight on their advantages comparing themselves to others. There is no global nor automatic approach taking advantage of all of them. But the better data are prepared, the better results will be, and the faster mining algorithms will work. Previously in [4], we proposed a simple but efficient approach to transform input data into a set of intervals (also called bins, clusters, classes, *etc.*). On which we apply, in a further step, specific mining algorithms (correlation rules, *etc.*). The reasons that decided us to reconsider previous works are: (i) To improve the outliers' detection with regard to the data distribution (normal or not), (ii) To reduce the number of input parameters, and thus (iii) To propose an automatic choice of the best discretization method. Finally, regarding implementation, we merge the three tasks into a single one, and carry out experiments.

This paper is organized as follows: Section II presents general aspects of data preparation. Section III and Section IV are dedicated to outlier detection and to discretization methods respectively. Each section is composed of two parts: (i) related work, and (ii) our approach for improving it. In Section V, we show the results of first experiments. Last Section summarizes our contribution, and outlines some research perspectives.

II. Data Preparation

Raw input data must be prepared in any Knowledge and Discovery in Databases (KDD) system previous to the mining step. There are two main reasons for this:

- If each value of each column is considered as a single item, there will be a combinatorial explosion of the search space, and thus very large reponse times;
- We cannot expect this task to be performed by hand because manual cleaning of data is time consuming and subject to many errors.

Generally, this step is divided into two tasks: (i) Preprocessing, and (ii) Transformation(s).

A. Preprocessing

Preprocessing consists in reducing the data structure by eliminating columns and rows of low significance [5].

a) Basic Column Elimination: Elimination of a column can be the result of, for example in the microelectronic industry, a sensor dysfunction, or the occurrence of a maintenance step; this implies that the sensor cannot transmit its values to the database. As a consequence, the associated column will contain many null/default values and must then be deleted from the input file. Elimination should be performed by using the Maximum Null Values ($MaxNV$) threshold. Furthermore, sometimes several sensors measure the same information, what produces identical columns in the database. In such a case, only a single column should be kept.

b) Elimination of Concentrated Data and Outliers: We first turn our attention to inconsistent values, such as "outliers" in noisy columns. Detection should be performed through another threshold (a convenient value of p when using the standardization method, see Section III-A1). Found outliers are eliminated by forcing their values to Null. Another technique is to eliminate the columns that have a small standard deviation (threshold $MinStd$): Since their values are almost the same, we can assume that they do not have a significant impact on results; but their presence pollutes the search space and reduces response times. Similarly, the number of Distinct Values in each column should be bounded by the minimum ($MinDV$) and the maximum ($MaxDV$) values allowed.

B. Transformation

c) Data Normalization: This step is optional. It translates numeric values into a set of values between 0 and 1. Standardizing data simplifies their classification.

d) Discretization: Discrete values deal with intervals of values, which are more concise to represent knowledge, so that they are easier to use and also more comprehensive than continuous values. Many discretization algorithms (*see* Section IV-A) have been proposed over the years for this. The number of used intervals ($NbBins$) as well as the selected discretization method among those available are here again parameters of the current step.

 ISBN: 978-1-61208-415-2

e) Pruning step: When the occurrence frequency of an interval is less than a given threshold ($MinSup$), then it is removed from the set of bins. If no bin remains in a column, then that column is entirely removed.

The presented thresholds/parameters are the ones we use for data preparation. In previous works, their values were fixed inside of a configuration file read by our software at setup. The objective of this work is to automatically determine most of these variables without information loss. Focus is set on outlier and discretization management.

III. Detecting Outliers

An outlier is an atypical or erroneous value corresponding to a false measurement, a calculation mistake, an unwritten input, *etc.* Outlier detection is an uncontrolled problem because extreme values deviate too greatly from the rest of the data. In other words, they are associated with a significant deviation from the other observations [3]. In this section, we present some outlier detection methods and focus on the detection of outliers in the case of uni-variate data.

The following notations are used to describe outliers: X is a numeric attribute of a database relation, and is increasingly ordered. x is an arbitrary value, X_i is the i^{th} value, N the size of X, σ its standard deviation, μ its mean, and s a central tendency parameter (variance, inter-quartile range, *etc.*). X_1 and X_n are the minimum and the maximum values of X respectively. p is an arbitrary probability, and k is a parameter specified by the user, or computed by the system.

A. Related Work

In this section, we summarize four of the principal univariate outlier detection methods.

1) Elimination after standardizing the distribution: This is the most conventional cleaning method [3]. It consists in taking into account μ and σ to determine the limits beyond which aberrant values will be eliminated. For an arbitrary distribution, the Bienaymé-Tchebyshev inequality specifies that the probability that the absolute deviation between a variable and its average is greater than p is less than or equal to $\frac{1}{p^2}$:

$$P(\left|\frac{x-\mu}{\sigma}\right| \geq p) \leq \frac{1}{p^2} \quad (1)$$

The idea is to set a threshold probability as a function of μ and σ above which we accept values as non-outliers. For example, with $p = 4.47$, the probability that x, satisfying $\left|\frac{x-\mu}{\sigma}\right| \geq p$, is an outlier is bounded by 0.05.

2) Algebraic method: This general detection method, presented in [6], uses the relative distance of a point to the "center" of the distribution: $d_i = \frac{|X_i-\mu|}{\sigma}$. Outliers are detected outside of the interval $[\mu - kQ_1, \mu + kQ_3]$, where k is generally set between 1.5 and 3. Q_1 and Q_3 are the first and the third quartiles respectively.

3) Box plot: This method, attributed to Tukey [7], does not make any assumption on how the data are distributed. It is based on the difference between quartiles Q_1 and Q_3, and distinguishes between two categories of extreme values determined outside the lower bound (LB) and the upper bound (UB):

$$\begin{cases} LB = Q_1 - k \times (Q_3 - Q_1) \\ UB = Q_3 + k \times (Q_3 - Q_1) \end{cases} \quad (2)$$

4) Grubbs' test: Grubbs' method, presented in [8], is a statistical test for lower or higher abnormal data. It uses the difference between the average and the extreme values of the sample. The test is based on the assumption that the data are normally distributed. The maximum and minimum values are tested, which allows one to decide if any of these values is aberrant. The statistic used is $T = max(\frac{X_N-\mu}{\sigma}, \frac{\mu-X_1}{\sigma})$. The test is based on two hypotheses:

- Hypothesis H_0: The tested value is not an outlier.
- Hypothesis H_1: The tested value is an outlier.

Hypothesis H_0 is rejected at significance level α if:

$$T > \frac{N-1}{\sqrt{n}} \sqrt{\frac{\beta}{n-2\beta}} \quad (3)$$

where $\beta = t_{\alpha/(2n),n-2}$ is the quartile of order $\alpha/(2n)$ of the Student distribution with $n-2$ degrees of freedom.

B. An Original Method for Outlier Detection

Many existing outlier detection methods assume that the distribution of data is normal. However, we observed that, in reality, many samples have asymmetric and/or multimodal distributions; the use of these methods will then have a significant influence on the mining step. Therefore, we should process each distribution using an appropriated method. The considered approach consists in eliminating outliers in each column based on the normality of data, in order to minimize the risk of eliminating normal values.

Firstly, the Kolmogorov-Smirnov test presented in [9] is applied in order to determine whether the distribution is normal or not. Secondly, if the variable is normally distributed, then the Grubbs' test is used at a significance level of 5%. This test gives experimentally better results than the algebraic approach. Otherwise, the *Box plot* method is employed with parameter k set to 3 in order to not to be too exhaustive toward outlier detection. Figure 1 summarizes the process we chose for detecting and deleting outliers.

In the previous versions of our software, we used the simple standardization method with p set as an input parameter. With this new approach, no input parameter remains. We obtained moreover an improvement of 2% in the detection of true positive or false negative outliers.

IV. Discretization Methods

Discretization of an attribute consists in finding $NbBins$ disjoint intervals that will further represent it in an efficient way. The final objective of discretization methods is to ensure that the mining part of the KDD process generates substantial results. In our approach, we only employ direct discretization methods in which $NbBins$ must be known in advance (and be the same for every column of the input data). $NbBins$ was initially a parameter fixed by the end-user. The literature proposes several formulas as an alternative (Rooks-Carruthers, Huntsberger, Scott, *etc.*) for computing such a

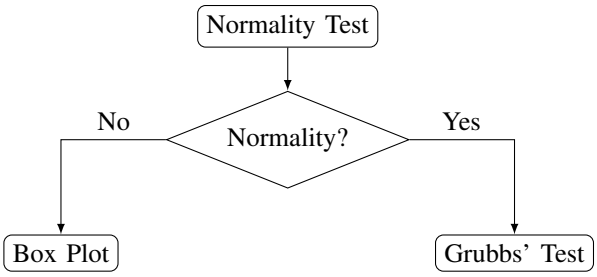


Figure 1. Main tests used in our outlier detection process.

number. Therefore, we switched to the Huntsberger formula, the most fitting from a theoretical point of view [10], and given by: $1 + 3.3 \times log_{10}(N)$.

A. Related Work

In this section, we only highlight the final discretization methods kept for this work. This is because the other tested methods have not revealed themselves to be as efficient as expected (such as Embedded Means Discretization), or are not a worthy alternative (such as Quantiles based Discretization) to the ones presented. The methods we use are: Equal Width Discretization (EWD), Equal Frequency-Jenks Discretization (EFD-Jenks), AVerage and STandard deviation based discretization (AVST), and K-Means (KMEANS). These methods, which are unsupervised [11] and static [12], have been widely discussed in the literature: See for example [10] for EWD and AVST, [13] for EFD-Jenks, or [2] and [14] for KMEANS. For these reasons, we only summarize their main characteristics and their field of applicability in Table I.

TABLE I. SUMMARY OF THE DISCRETIZATION METHODS USED.

Method	**Principle**	**Applicability**
EWD	This simple to implement method creates intervals of equal width.	The approach cannot be applied to asymmetric or multimodal distributions.
EFD-Jenks	Jenks' method provides classes with, if possible, the same number of values, while minimizing internal variance of intervals.	The method is effective from all statistical points of view but presents some complexity in the generation of the bins.
AVST	Bins are symmetrically centered around the mean and have a width equal to the standard deviation.	Intended only for normally distributed datasets.
KMEANS	Based on the Euclidean distance, this method determines a partition minimizing the quadratic error between the mean and the points of each interval.	The disadvantage of this method is its exponential complexity, so computation time can be long. It is applicable to each form of distribution.

Let us underline that the upper limit fixed to the number of intervals to use is not always reached, depending on the applied discretization method. Thus, EFD-Jenks and KMEANS generate most of the times less than $NbBins$ bins.

Example 1: Let us consider the numeric attribute $SX = \{4.04, 5.13, 5.93, 6.81, 7.42, 9.26, 15.34, 17.89, 19.42, 24.40, 25.46, 26.37\}$. SX contains 12 values, so by applying the Huntsberger's formula, if we aim to discretize this set, we have to use 4 bins.

Table II shows the bins obtained by applying all the discretization methods proposed in Table I. Table III shows the number of values of SX belonging to each bin associated to every discretization method.

TABLE II. SET OF BINS ASSOCIATED TO SAMPLE SX.

Method	Bin_1	Bin_2	Bin_3	Bin_4
EWD	[4.04, 9.62[	[9.62, 15.21[	[15.21, 20.79[	[20.79, 26.37]
EFD-Jenks	[4.04; 5.94]	]5.94, 9.26]	]9.26, 19.42]	]19.42, 26.37]
AVST	[4.04; 5.53[	[5.53, 13.65[	[13.65, 21.78[	[21.78, 26.37]
KMEANS	[4.04; 6.37[	[6.37, 12.3[	[12.3, 22.95[	[22.95, 26.37]

TABLE III. POPULATION OF EACH BIN OF SAMPLE SX.

Method	Bin_1	Bin_2	Bin_3	Bin_4
EWD	6	0	3	3
EFD-Jenks	3	3	3	3
AVST	2	4	4	2
KMEANS	3	3	4	2

As it is easy to understand, we cannot find two discretization methods producing the same set of bins. As a consequence, the distribution of the values of SX is different depending on the method used.

B. Discretization Methods and Statistical Characteristics

When attempting to find the most appropriate discretization method for a column, what is important is not the law followed by its distribution, but the shape of its density function. This is why we first perform a descriptive analysis of the data in order to characterize, and finally to classify, each column according to normal, uniform, symmetric, antisymmetric or multimodal distributions. This is done in order to determine what discretization method(s) may apply. Let us underline that the proposed tests have to be performed in the given order:

1) We use the Kernel method presented in [15] to characterize multimodal distributions. The method is based on estimating the density function of the sample by building a continuous function, and then calculating the number of peaks using its second derivative. It involves building a continuous density function, which allows us to approximate automatically the shape of the distribution. The multimodal distributions are those which have a number of peaks greater than 1.
2) To characterize antisymmetric distributions in a next step, we use the Skewness, noted γ_3:

$$\gamma_3 = E[(\frac{X - \mu}{\sigma})^3] \quad (4)$$

The distribution is symmetric if $\gamma_3 = 0$. Practically, this rule is too exhaustive, so we relaxed it by imposing limits around 0 to set a fairly tolerant rule which allows us to decide whether a distribution is considered antisymmetric or not. The associated method is based on a statistical test. The null hypothesis is that the distribution is symmetric.
Consider the statistic: $T_{Skew} = \frac{N}{6}(\gamma_3^2)$. Under the null hypothesis, T_{Skew} follows a law of χ^2 with one

degree of freedom. In this case, the distribution is antisymmetric if $\alpha = 5\%$ if $T_{Skew} > 3.8415$.

3) We use then the normalized Kurtosis, noted γ_2, to measure the peakedness of the distribution or the grouping of probability densities around the average, compared with the normal distribution. When γ_2 is close to zero, the distribution has a normalized peakedness:

$$\gamma_2 = E[(\frac{X-\mu}{\sigma})^4] - 3 \quad (5)$$

A statistical test is used again to automatically decide whether the distribution has normalized peakedness or not. The null hypothesis is that the distribution has a normalized peakedness.

Consider the statistic: $T_{Kurto} = \frac{N}{6}(\frac{\gamma_2^2}{4})$. Under the null hypothesis, T_{Kurto} follows a law of χ^2 with one degree of freedom. The null hypothesis is rejected at level of significance $\alpha = 0.05$ if $T_{Kurto} > 6.6349$.

4) To characterize normal or uniform distributions, we use the Kolmogorov-Smirnov test, which can be used to compare the empirical functions of two samples if they have the same distribution.

These four successive tests allow us to characterize the shape of the (density function of the) distribution of every column. Combined with the main characteristics of the discretization methods presented in the last section, we get Table IV: This summarizes which discretization method(s) can be invoked depending on specific column statistics.

TABLE IV. APPLICABILITY OF DISCRETIZATION METHODS DEPENDING ON THE DISTRIBUTION'S SHAPE.

	Normal	Uniform	Symm-etric	Antisym-metric	Multimodal
EWD	*	*	*		
EFD-Jenks	*	*	*	*	*
AVST	*				
KMEANS	*	*	*	*	*

Example 2: Continuing Example 1, the Kernel Density Estimation method [15] is used to build the density function of sample SX (*cf.* Figure 2).

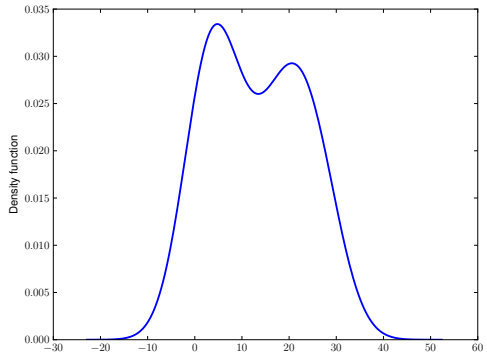


Figure 2. Density function of sample SX using Kernel Density Estimation.

As we can see, the density function has two modes, is almost symmetric and normal. Since the density function is multimodal, we should stop at this point. But as shown in Table IV, only EFD-Jenks and KMEANS produce interesting results according to our proposal. For the need of this example, let us perform the other tests. Since $\gamma_3 = -0.05$, the distribution is almost symmetric. As mentioned in (2), it depends on the threshold fixed if we consider that the distribution is symmetric or not. The distribution is not antisymmetric because $T_{Skew} = 0.005$. The distribution is not uniform since $\gamma_2 = -1.9$. As a consequence, $T_{Kurto} = 1.805$, and we have to reject the uniformity test. The Kolmogorov-Smirnov test results indicate that the probability that the distribution follows a normal law is 86.9% with $\alpha = 0.05$. Here again, accepting or rejecting the fact that we can consider if the distribution is normal or not depends on the fixed threshold.

C. Multi-criteria Approach for Finding the Most Appropriate Discretization Method

Discretization must keep the initial statistical characteristics so as the homogeneity of the intervals, and reduce the size of the final data produced. So, the discretization objectives are many and contradictory. For this reason, we chose a multi-criteria analysis to evaluate the available applicable methods of discretization. We use three criteria:

- The entropy H measures the uniformity of intervals. The higher the entropy, the more the discretization is adequate from the viewpoint of the number of elements in each interval:

$$H = -\sum_{i=1}^{NbBins} p_i \log_2(p_i) \quad (6)$$

where p_i is the number of points of interval i divided by the total number of points (N), and $NbBins$ is the number of intervals. The maximum of H is computed by discretizing the attribute into $NbBins$ intervals with the same number of elements. In this case, H reduces to $log_2(NbBins)$.

- The index of variance J, introduced in [10], measures the interclass variances proportionally to the total variance. The closer the index is to 1, the more homogeneous the discretization is:

$$J = 1 - \frac{\text{Intra-intervals variance}}{\text{Total variance}}$$

- Finally, the stability S corresponds to the maximum distance between the distribution functions before and after discretization. Let F_1 and F_2 be the attribute distribution functions before and after discretization respectively:

$$S = sup_x(\left|F_1(x) - F_2(x)\right|) \quad (7)$$

The objective is to find solutions that present a compromise between the various performance measures. The evaluation of these methods should be done automatically, so we are in the category of *a priori* approaches where the decision-maker intervenes just before the evaluation process step.

Aggregation methods are among the most widely used methods in multi-criteria analysis. The principle is to reduce

to a unique criterion problem. In this category, the weighted sum method involves building a unique criterion function by associating a weight to each criterion [16][17]. This method is limited by the choice of the weight, and requires comparable criteria. The method of inequality constraints is to maximize a single criterion by adding constraints to the values of the other criteria [18]. The disadvantage of this method is the choice of the thresholds of the added constraints.

In our case, the alternatives are the 4 methods of discretization, and we discretize automatically columns separately, so the implementation facility is important in our approach. Hence the interest in using the aggregation method by reducing it to a unique criterion problem, by choosing the method that minimizes the Euclidean distance from the target point $(H = log_2(NbBins),\ J = 1,\ S = 0)$.

Definition 1: Let D be an arbitrary discretization method, and V_D a measure of segmentation quality using the proposed multi-criteria analysis:

$$V_D = \sqrt{(H_D - log_2(NbBins))^2 + (J_D - 1)^2 + S_D^2} \quad (8)$$

The following proposition is the main result of this article: It indicates how we chose the most appropriate discretization method among all the available ones.

Proposition 1: Let DM be a set of discretization methods; the one, noted $\mathbb{D}$, that minimizes V_{D} (*see* equitation 8), $\forall D \in \{DM\}$, is the best discretization method.

Corollary 1: The most appropriate discretization method $\mathbb{D}$ can be obtained as follows:

$$\mathbb{D} = argmin(\{V_{\mathrm{D}}, \forall D \in \{DM\}\}) \quad (9)$$

As a result of corollary 1, we propose the MAD (Multi-criteria Analysis for finding the best Discretization method) algorithm, see Figure 3.

Input: X set of numeric values to discretize, DM set of discretization methods applicable
Output: best discretization method for X
1: **for each** method $D \in$ DM **do**
2: Compute V_D
3: **end for**
4: **return** $argmin(V)$

Figure 3. MAD: Multi-criteria Analysis for Discretization

Example 3: Continuing Example 1, Table V shows the evaluation results for all the discretization methods at disposal. Let us underline that for the need of our example, all the values are computed for every discretization method, and not only for the ones which should have been selected after the step proposed in Section IV-B (*cf.* Table IV). The results show that EFD-Jenks and KMEANS are the two methods that obtain the lowest values for V_D. The values got by the EWD and AVST methods are the worst: This is consistent with our optimization proposed in table IV, since the sample distribution is multimodal.

TABLE V. EVALUATION OF DISCRETIZATION METHODS.

	H	J	S	V_{DM}
EWD	1.5	0.972	0.25	0.313
EFD-Jenks	2	0.985	0.167	0.028
AVST	1.92	0.741	0.167	0.101
KMEANS	1.95	0.972	0.167	0.031

V. EXPERIMENTAL ANALYSIS

In this section, we present some experimental results by evaluating three samples. We decided to implement it using the MineCor KDD Software [4], but it could have been with another one (R Project, Tanagra, *etc.*) Sample_1 is a randomly generated file that contains heterogeneous values. Sample_2 and Sample_3 correspond to real data representing measurements provided by a microelectronics manufacturer (STMicroelectronics) after completion of the manufacturing process. Table VI sums up the characteristics of the samples.

TABLE VI. CHARACTERISTICS OF THE DATABASES USED.

Sample	Number of columns	Number of rows	Type
Sample_1	9	468	generated
Sample_2	7	727	real
Sample_3	1281	296	real

Figures 4 and 5 summarize respectively the evaluation of the methods used on the two first samples.

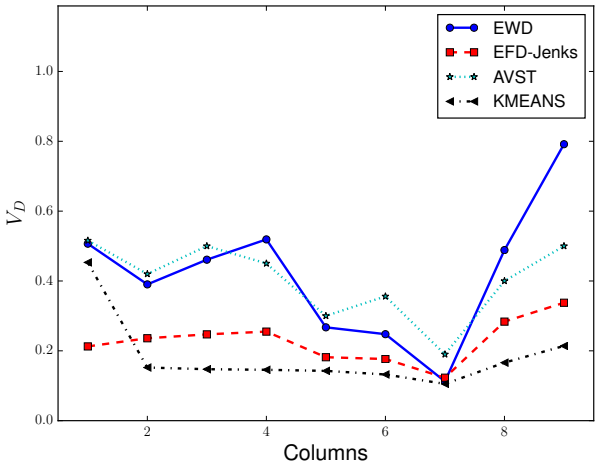


Figure 4. DM comparison on Sample_1's columns.

For the Sample_1 evaluation shown graphically in Figure 4, the columns studied have relatively dispersed, asymmetric and multimodal distributions. "Best" discretizations are provided by EFD-Jenks and KMEANS methods. We note also that the EWD method is fast, and sometimes demonstrates good performances in comparison with the EFD-Jenks or KMEANS methods.

For Sample_2 attributes, which have symmetric and normal distributions, the evaluation on Figure 5 shows that the EFD-Jenks method provides generally the best results. The

 ISBN: 978-1-61208-415-2

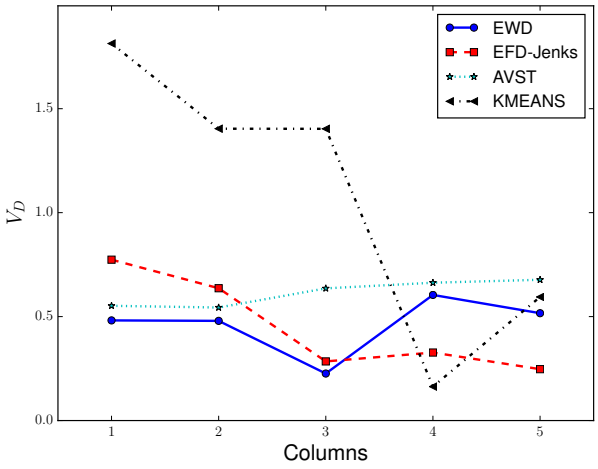


Figure 5. DM comparison on Sample$_2$'s columns.

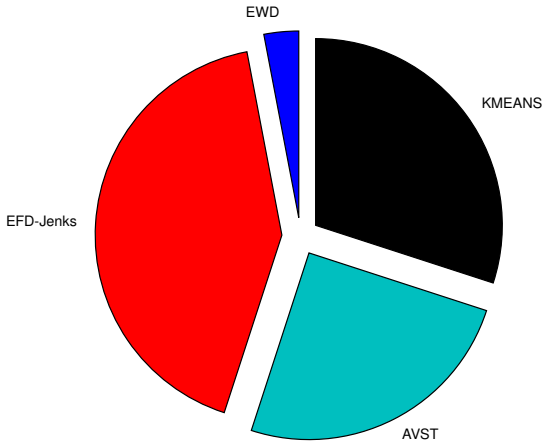


Figure 6. Selected Discretization Method.

KMEANS method is unstable for these types of distributions, but sometimes provides the best discretization.

Finally, Figure 6 summarizes our approach: We have tested it over each column of each dataset. Any of the available methods is selected at least once in the dataset of the three proposed samples, which enforces our approach. As expected, EFD-Jenks is the method that is the most often kept by our software ($\simeq$ 42%). AVST and KMEANS are selected approximately 30% each. EWD is only selected a very few times (less than 2%).

VI. Conclusion and Future Work

In this paper, we presented a new approach for automatic data preparation implementable in most of KDD systems. This step is generally split into two sub-steps: (i) detecting and eliminating the outliers, and (ii) applying a discretization method in order to transform any column into a set of clusters. In this article, we show that outliers' detection is depending on if data distribution is normal or not. As a consequence, we do not have to apply the same pruning method (Box plot *vs.* Grubb's test). Moreover, when trying to find the most appropriate discretization method, what is important is not the law followed by the column, but the shape of its density function. That is why we propose an automatic choice for finding the best discretization method based on a multi-criteria approach. Experimental evaluations done on real and synthetic data validate our work, showing that it is not always the very same discretization method that is the best: Each method has its strengths and drawbacks.

For future works, we aim to experimentally validate the relationship between the distribution shape and the applicability of used methods, to add other discretization methods (Khiops, Chimerge, Entropy Minimization Discretization, *etc.*) to our system, to parallelize our work using the latest functionalities of multicore programming, and to measure the impact of the data preparation step on the results of some mining algorithms (association rules, correlation rules, *etc.*).

References

[1] D. Pyle, Data Preparation for Data Mining. Morgan Kaufmann, 1999.

[2] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 24, no. 7, 2002, pp. 881–892.

[3] C. Aggarwal and P. Yu, "Outlier detection for high dimensional data," in SIGMOD Conference, S. Mehrotra and T. K. Sellis, Eds. ACM, 2001, pp. 37–46.

[4] C. Ernst and A. Casali, "Data preparation in the minecor kdd framework," in IMMM 2011, The First International Conference on Advances in Information Mining and Management, 2011, pp. 16–22.

[5] O. Stepankova, P. Aubrecht, Z. Kouba, and P. Miksovsky, "Preprocessing for data mining and decision support," in Data Mining and Decision Support: Integration and Collaboration, K. A. Publishers, Ed., 2003, pp. 107–117.

[6] M. Grun-Rehomme, O. Vasechko et al., "Méthodes de détection des unités atypiques: Cas des enquêtes structurelles ukrainiennes," in 42èmes Journées de Statistique, 2010.

[7] J. W. Tukey, "Exploratory data analysis. 1977," Massachusetts: Addison-Wesley, 1976.

[8] F. E. Grubbs, "Procedures for detecting outlying observations in samples," Technometrics, vol. 11, no. 1, 1969, pp. 1–21.

[9] H. W. Lilliefors, "On the kolmogorov-smirnov test for normality with mean and variance unknown," Journal of the American Statistical Association, vol. 62, no. 318, 1967, pp. 399–402.

[10] C. Cauvin, F. Escobar, and A. Serradj, Cartographie thématique. 3. Méthodes quantitatives et transformations attributaires. Lavoisier, 2008.

[11] I. Kononenko and S. J. Hong, "Attribute selection for modelling," Future Generation Computer Systems, vol. 13, no. 2, 1997, pp. 181–195.

[12] S. Kotsiantis and D. Kanellopoulos, "Discretization techniques: A recent survey," GESTS International Transactions on Computer Science and Engineering, vol. 32, no. 1, 2006, pp. 47–58.

[13] U. of Kansas. Dept. of Geography and G. Jenks, Optimal data classification for choropleth maps, 1977.

[14] A. K. Jain, "Data clustering: 50 years beyond k-means," Pattern Recognition Letters, vol. 31, no. 8, 2010, pp. 651–666.

[15] B. W. Silverman, Density estimation for statistics and data analysis. CRC press, 1986, vol. 26.

[16] P. M. Pardalos, Y. Siskos, and C. Zopounidis, Advances in multicriteria analysis. Springer, 1995.

[17] B. Roy and P. Vincke, "Multicriteria analysis: survey and new directions," European Journal of Operational Research, vol. 8, no. 3, 1981, pp. 207–218.

[18] M. Chilali, "Méthodes lmi pour l'analyse et la synthèse multi-critère." Ph.D. dissertation, 1996.

 ISBN: 978-1-61208-415-2

Towards Predictive Policing: Knowledge-based Monitoring of Social Networks

Michael Spranger, Florian Heinke, Steffen Grunert and Dirk Labudde
University of Applied Sciences Mittweida
Mittweida, Germany
Email: {*name.surname*}@hs-mittweida.de

***Abstract*—Increasing the resilience of the society against disorders, such as disasters, attacks or threatening groups, is a major challenge. Recent events highlight the importance of a resilient society and steps which are required to be taken in resilience engineering. *A priori* the optimal way to handle such adverse events is to prevent them, or at least provide appropriate courses of preparation. The essential requirement for every kind of preparation is information about relevant upcoming events. Such information can be gained for example from social networks and can form the basis for a long-term and short-term strategic planning by security forces. For that purpose, we here propose an application framework for knowledge-based social network monitoring, which aims at predicting short-term activities, as well as the long-term development of potentially dangerous groups. In this work, a theoretical outline of this approach is given and discussed.**

Keywords–forensic; text processing; resilience engineering

I. INTRODUCTION

The representation and the communication via the Internet, especially in social networks, have become a standard not only for individuals, companies and organizations but also for political groups or gangs using these platforms for planning, appointing and conducting criminal offences [1], [2]. Large events with a relatively large degree of group dynamics, like sport events, demonstrations or festivals, require a high expenditure of staff on the side of the security forces because of unpredictability and uncertainty of associated dynamics. For example, to secure the soccer events in 2014 in Germany approximately two million working hours of police officers were necessary [3]. In order to support decision makers, we outline an application framework for monitoring cliques and groups in social networks, which can be key elements in the emergence of critical events. The monitoring process is facilitated by means of employing general domain-specific endangerer profiles. Such a profile can be deduced from a set of social network sites of known endangerers or perpetrators (in the strict sense). Identifying suspicious activities is realized by group recommendation classifiers.
The following section is structured according to the steps required to generate the proposed framework. First, aspects of ontology definition are outlined, followed by discussions on endangerer profile generation and classifier training. Finally, monitoring strategies are proposed.

II. PROPOSAL

The proposed application framework enables decision makers of security forces to identify threat hot-spots. In this way, they are able to control their human resources. In order to support long-term resource planning, The second aim is to predict the long-term development of groups that pose a threat. The process pipeline consists of three parts:

1) modelling the threat ontology
2) train the general domain-specific endangerers profile
3) monitoring all matching social network sites and calculate a long-term and short-term threat score

A. Threat Ontology

The term ontology in a common understanding means a formal and explicit specification of a common conceptualization. In particular, it is defined as a set of common classified terms and symbols referred to a syntax, and a network of associate relations [4]. Similar to the crime ontology we proposed in recent work [5], an ontology can be used for modelling a complex threat assessment. In this way, knowledge of decision makers is introduced and can be used for extracting semantic information from posts and comments of social network's profiles. In particular, the works of Wimalasuriya and Dou [6], Embley [7] and Maedche [8], show that the use of ontologies is suitable for assisting the extraction of semantic units, as well as their visualization and structures such processes very well.

B. Endangerer Profile

In order to distinguish profiles of interest regarding to a certain threat, a general profile needs to be modelled. Recent work [9], [10] has shown that feature vectors derived from social network profiles are suitable for generating group recommendations. In a similar way, a general classifier can be trained based on the social network profiles of known persons associated with a special threat. For example, Facebook profiles of known hooligans of a specific soccer club can be used to train classifiers that are able to identify social activity of hooligans and peers in social networks.
The generation process is divided into three parts depicted in Figure 1.

C. Monitoring Activities

Once a profile is generated and the threat specific ontology is defined, the social network monitoring can be conducted. At this point a multi-level, information extraction process aims at instantiating the ontology using textual information, like posts and comments. An example of how such a process can be structured is given by Spranger and Labudde [5]. Further text analysis steps, like sentiment analysis (see the discussions

 ISBN: 978-1-61208-415-2

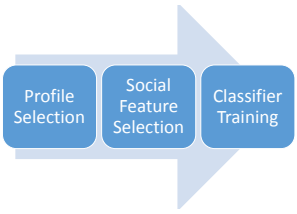


Figure 1. The process of deriving a threat specific general profile.

given in [11] and [12] for details) can complete the instantiated model in different ways. As a short-term benefit, a score can be computed for various time points, signalling whether a threatening event regarding to the specific profile and ontology is directly pointing to a specific location and time frame. These results can be applied to a map to localize short-term hot-spots in terms of security and their dynamics as discussed by Davies and Bishop [13].

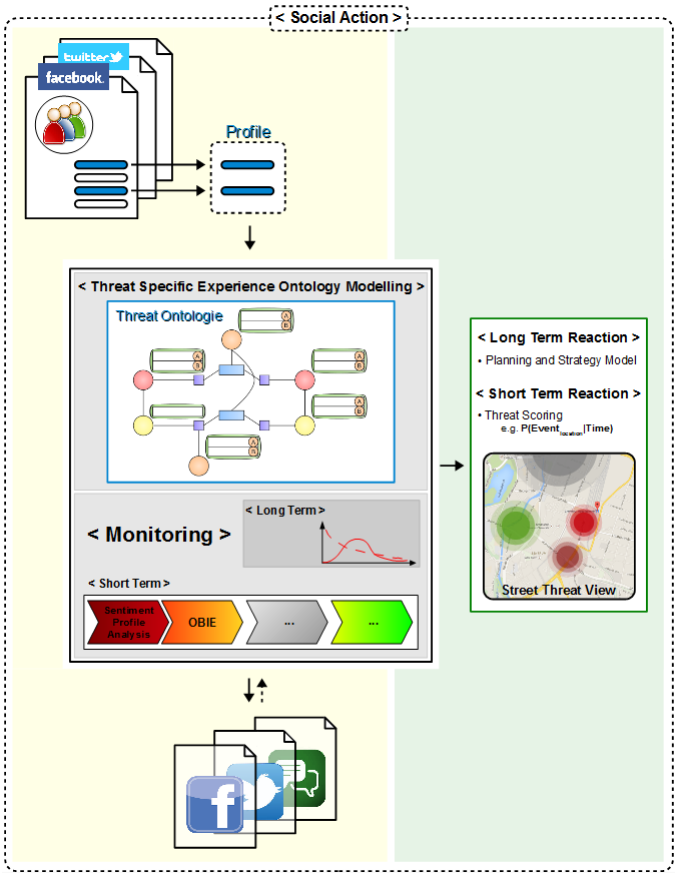


Figure 2. The proposed system. The central, expert-modelled threat-specific ontology describes the environment of a special threat. A general endangerer profile completes the model. In the process the model is used to extract textual information from social network activities. Different scoring functions allow the identification of threat hot-spots or can show the long-term evolution of groups and cliques.

In the age of Big Data and algorithms handling such amounts of information, deducing long term developments of such groups and dynamics is at its early stage. Methodological concepts widely used in modelling complex relations (as for instance systems biology) can be directly transferred to the field of resilience engineering. Especially, employing generic mathematical models to social networks has become computationally feasible, but requires further research. For example, epidemiological models can be efficiently applied to study long term evolutions of groups and sub-networks (see [14]) and study the information transfer between them. Thus, generating valid models and derive predictions from them can be of great value, for instance, in planning personnel and staff demands.

REFERENCES

[1] ITU. Number of worldwide internet users from 2000 to 2014 (in millions). statista. [Online]. Available: http://www.statista.com/statistics/273018/number-of-internet-users-worldwide/ (2015)

[2] eMarketer & American Marketing Association. Number of social network users worldwide from 2010 to 2018 (in billions). statista. [Online]. Available: http://www.statista.com/statistics/278414/number-of-worldwide-social-network-users (2015)

[3] ZIS. Jahresbericht 2013/14. Zentrale Informationsstelle Sporteinsätze. [Online]. Available: http://www.polizei-nrw.de/media/Dokumente/Behoerden/LZPD/ZIS_Jahresbericht_2013_14.pdf (2014)

[4] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," in Formal Ontology in Conceptual Analysis and Knowledge Representation, N. Guarino and R. Poli, Eds. Kluwer Academic Publishers, 1993.

[5] M. Spranger and D. Labudde, "Towards establishing an expert system for forensic text analysis," International Journal on Advances in Intelligent Systems, vol. 7, no. 1/2, 2014, pp. 247–256.

[6] D. C. Wimalasuriya and D. Dou, "Ontology-based information extraction: An introduction and a survey of current approaches," Journal of Information Science, vol. 36, no. 3, 2010, pp. 306–323.

[7] D. W. Embley, "Toward semantic understanding: an approach based on information extraction ontologies," in Proceedings of the 15th Australasian database conference - Volume 27, ser. ADC '04. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2004, pp. 3–12.

[8] A. Maedche, G. Neumann, and S. Staab, "Bootstrapping an ontology-based information extraction system," Studies In Fuzziness And Soft Computing, vol. 111, 2003, pp. 345–362.

[9] M. Manca, L. Boratto, and S. Carta, "Producing friend recommendations in a social bookmarking system by mining users content," in Proc. 3rd. International Conference on Advances in Information Mining and Management, IARIA. ThinkMind Library, 2013, p. 59 to 64.

[10] M. Cheung and J. She, "Bag-of-features tagging approach for a better recommendation with social big data," in Proc. 4th. International Conference on Advances in Information Mining and Management, IARIA. ThinkMind Library, 2014, p. 83 to 88.

[11] S. M. Mohammad, S. Kiritchenko, and X. Zhu, "Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets," in Proceedings of the Second Joint Conference on Lexical and Computational Semantics (SEMSTAR'13), 2013.

[12] X. Wan, "Co-training for cross-lingual sentiment classification," in Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1. Association for Computational Linguistics, 2009, pp. 235–243.

[13] T. Davies and S. Bishop, "Modelling patterns of burglary on street networks," Crime Science, vol. 2, no. 1, 2013, p. 10.

[14] J. Cannarella and J. A. Spechler, "Epidemiological modeling of online social network dynamics," CoRR, vol. abs/1401.4208, 2014.

 ISBN: 978-1-61208-415-2

Real-Time Partition of Streamed Graphs for Data Mining over Large Scale Data

Víctor Medel and Unai Arronategui
Escuela de Ingeniería y Arquitectura
University of Zaragoza
Zaragoza, Spain
{vmedel, unai@unizar.es}

***Abstract*—Mining data in real-time from large graphs requires a lot of memory to obtain a good distribution of information. Current state of the art solutions for streamed graphs are not scalable and they work with a single stream source. We propose a new reduced memory model to partition large graphs over big streams to improve mining algorithms. The aim of our work is to give support to data mining algorithms over large-scale structured data (e.g., Web structure, social networks) to minimise communication among partitions. In our architecture, the incoming graph elements are sampled to reduce total memory usage and the information in each partitioner is updated in a feedback scheme to allow multiple entry points. We have made experimentation with real-world graphs and we have discussed about the suitability of different sampling strategies depending on the graph structure. In addition, we have executed the PageRank algorithm over the partitioned graph, in order to measure the influence of the partition in the execution of a mining algorithm.**

Keywords–Big Graphs; Data Streaming; Graph Partition; Sampling.

I. INTRODUCTION

Mining from social networks, from Wikipedia or from World Wide Web compels to deal with a huge amount of information modelled by a graph. Information is continuously growing, so it has to be processed as is generated. The data stream paradigm [1] fits well with these kind of applications. As we want a real-time processing, the resource needs are very huge.

The underlaying graph is so large that it cannot be stored in a single machine, thus it has to be distributed among several machines before we can make some analytics over it. For example, Yahoo! Dataset [2] is 120 GB in size, and a web network graph of 50 billion vertices and 1 trillion edges, like the one used by Google in Pregel experimentation [3], needs 25 TB of free storage space.

Typical graph analytics algorithms in these domains (e.g., PageRank [4], Community Discovering [5], Triangle Counting [6], [7]) need a lot of communication among vertices, so the number of edges among partitions (cutting edges) will condition network traffic and therefore execution time. In other words, the quality of the partition solution has a direct impact in the execution of task over the graph.

Partitioning a graph in a streaming scenario is a novel application with a few works [8],[9]. Proposed algorithms do not scale well. They make an intensive use of memory because they need to have knowledge of previous elements of the graph. In addition, they only consider a single stream source, consequently their incoming rate is bound by network capacity.

In this work, we focus on graph partition problem in a streaming environment with hard resource constraints, in order to guarantee real-time data management. We consider a single-pass streaming algorithm, with multiple stream sources and we reduce the total memory usage, to propose a scalable model. Over the obtained partitions, we have executed the PageRank algorithm to show the trade-off between used memory in partition phase and total time of the execution of an analytic. We have used the PageRank algorithm to compare our results as it has been the reference analytic used in previous works.

We have achieved our aims by sampling incoming elements and by updating, periodically, the information in each partitioner. Our model is high scalable, it is not bind by network capacity and it allows multiple streaming inputs.

The paper is structured in six parts, with this Introduction as first section. In Section II, we synopsise fundamental notions in graph partition problem in a streaming scenario. In Section III, we talk about the state of the art in the domain and in Section IV, we present our architecture. The analysis of our model is made in Section V. In Section VI, experimental results from a real scenario are shown. Finally, the conclusion and future directions of the research are presented in Section VII.

II. BACKGROUND

In this section, we present the fundamental notions used in this paper. We start showing how to model a graph in a streaming environment and the graph partition problem. At the end of the section, we analyse the requirements to guarantee real-time processing in data streams.

A. Graphs on data stream.

We consider that the graph arrives in a data stream way. A Data Stream A [10] is an ordered sequence of $a_1, a_2, ..a_n$ elements. In informal terms, the system has no control over the arriving model; streams are potentially unbound in size and once an element has been processed it cannot be retrieved.

We denote $G = (V, E)$ an undirected graph with vertex set $V = \{v_1, v_2, v_3, ...v_n\}$ and an edge set $E =$

 ISBN: 978-1-61208-415-2

$\{e_1, e_2, e_3, ..., e_m\}$. Note that *n* is the number of vertices, *m* the number of edges and $e_i = (v_j, v_k), \quad v_j, v_k \in V$.

A vertex graph stream, T, is a sequence of $t_1, t_2, ..., t_n$ where $t_j = (v_j, v_{j_1}, v_{j_2}, ..., v_{j_d})$, $v_j, v_{j_i} \in V$, $(v_j, v_{j_i}) \in E$ and $deg(v_j) = d$, for $i = 1, ..., d$ and for $j = 1, ..., n$.

Each tuple represents a vertex with its adjacency list. The size of a tuple depends on the degree of the vertex d, so processing time per tuple is variable. As we consider undirected graphs, each edge appears implicitly twice. In storage terms, the graph size is bound by $O(n + 4m)$.

Although in the general Data Stream model elements arrive in a random order; some specific models have been proposed for graph problems [9]. In Breadth First Search (BFS) model, one vertex of the graph is selected and, from that vertex, a breadth first search strategy is performed to generate the following vertices. A Depth First Search (DFS) strategy could be also performed. These orders have full sense in graph applications. For example, if a web crawler follows links with a BFS strategy, the elements of the graphs are generated with that order.

B. Graph partition problem.

Given a graph $G = (V, E)$, we define a k partition set P, where $P = \{S_1..S_k\}$ such as $S_i \subset G$ and $\bigcup_{i=1}^{k} S_i = G$. We define the graph partition problem as finding an optimal P^* such that for all possible partitions P such that $|P| = k$, $f(P^*) \geq f(P)$, for a determinate function.

Our objective is to obtain a partition set P which minimises the communication cost among partitions S_i and consequently processing time of a mining algorithm over the partition. Thus, f depends on the following metrics:

$$\lambda = \frac{\textit{number of cutting edges}}{\textit{total edges}} = \frac{\Lambda}{m} \tag{1}$$

$$\rho = \frac{Max\{|S_i|, \forall i \in \{1, ..., k\}\}}{\frac{n}{k}} \tag{2}$$

An edge $(v_i, v_j) \in E$ is a cutting edge for a k partition set P if $v_i \in S_q$ and $v_j \in S_r$, with $i, j \in \{i, ..., n\}$ and $q, r \in \{1, ..., k\}$ and $r \neq q$.

The λ parameter (equation (1)) gives the possible overhead of needed communication among partitions when graph processing tasks are executed. The ρ value (equation (2)) is the balanced factor of the solution partition set P. In the processing phase, having too disbalanced partitions might increase the processing time (some machines have to do a heavy process and others might be idles).

This problem is NP-Complete [11], so we have to consider approximations to the optimal solution. In [9] [8], light heuristics are used to compute an approximation of the best partition.

C. Real time streaming.

Incoming elements are processed as they arrive. We cannot store each new element because the stream is unbound. The partition algorithm cannot belong to $O(n)$ used memory algorithms. In an informal way, if we consider the graph partition problem, storing each incoming element would build the entire graph in a local or distributed memory. However, this is the same graph that has been partitioned. If the system needs to access the distributed memory for each vertex, additional time is lost. Consequently, if incoming elements have to be processed as they arrive, the total number of distributed partitioners must be increased. This increment would suppose to multiply the number of partitioners by a factor which would depend on the response time of the distributed memory.

Another possibility could be to query to partitions for each arriving vertex. This solution implies that incoming rate could be at most half of total network capacity, so we could not guarantee real-time. Moreover, partition algorithm cannot belong to $O(n)$ process time algorithms and it cannot do more than one pass over the stream.

The fact that we develop a single-pass algorithm with hard memory usage restrictions, makes our solution approximate. We compute a (ε, δ)-approximation of λ^* which means that $P[\lambda \leq (1 + \varepsilon)\lambda^*] \leq 1 - \delta$.

III. Related Work

Graph algorithms have been widely treated in literature. Graph streaming model has been described in [1] in a theoretical way. It represents the sequential access to graph elements instead of random access, due to the size of the graph. In this regard, several papers propose how to adapt graph algorithms to streaming paradigm [12] [13]. They take considerations about the complexity of different typical algorithms (triangle count [14], property checking, connectivity, etc.) and they calculate the required space bound and number of times an element of a stream is processed. In several works, they relax some data stream restrictions in order to obtain more flexible models: Semi-stream [1], W-Stream [15], Best-Order Stream [16], Sort-Stream, etc.

The main disadvantage of these works is that these restrictions cannot be made in an on-line environment with real-time considerations. As graph partition problem in a streaming environment is an NP-Complete problem [11], it is not feasible to compute the optimal solution, so we compute an approximated one. In [9] [8], some approximated solutions are obtained via light heuristics. The solution provided by Fennel [8] is quite good for real graphs. For some graphs, the obtained partition is as good as Metis [17], an offline partition algorithmn. In their experiments, the worst case is for *amazon0312* and they get a 6% more cutting edges. The problem of these kind of solutions is the linear size of the used memory, which makes difficult to scale the heuristic. In addition, it only considers a single input stream, which binds incoming rate.

IV. Proposed Model

We propose the decentralised architecture that is illustrated in Figure 1. We have uncoupled the different processing stages in order to distribute them. There are several loaders which continuously send elements to partitioners. They execute the partition algorithm to select the most suitable partition, and they send the element to that partition. The partition algorithm has to be simple, in computational terms, and it has to select

 ISBN: 978-1-61208-415-2

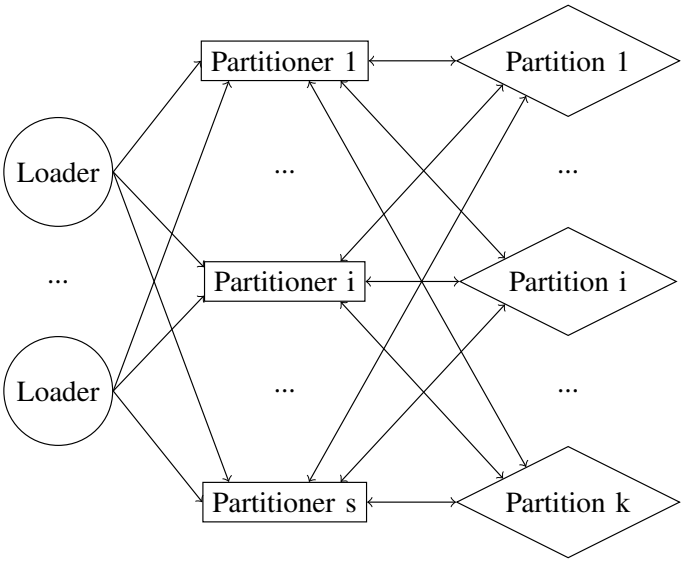


Figure 1. Proposed Architecture

the partition based on partial information. Its local information is updated by the partitions in a feedback scheme.

Memory size restriction has been solved sampling incoming vertices. We propose to group vertices in sets, in order to reduce total memory. We only have to store each sampling-set, which will be used by the algorithm to calculate the best partition. As the proposed framework is distributed, sampling functions cannot have knowledge of the entire graph. Each partitioner has to assign the same vertex to the same summary and all elements of a summary are assigned to the same partition. In order to maintain information consistency in each partitioner, each partition sends to them the set of sampling-sets stored periodically.

Let it be a graph $G = (V, E)$, a sampling size l and a sampled graph $G' = (\Psi, \Phi)$, where $\Psi = \{\Pi_1 ... \Pi_u\}$, $\Pi_i \subset V$, $\Phi = \{(\Pi_r, \Pi_q) \in \Psi,\ r, q = 1...u\}$ with $u = \frac{n}{l}$.

We define two surjectives sampling functions, g and h, where: $g : V \rightarrow 1...u$, $h : E \rightarrow \Phi$, such as:

i. $\forall v \in V,\ \exists \Pi_q \in \Psi \mid g(v) = q \Leftrightarrow v \in \Pi_q$
ii. $\forall i, j \in \{1, ..., n\},\ \forall q, r \in \{1, ..., u\},\ \forall v_i, v_j \in V,$ $\exists \Pi_r, \Pi_q \mid g(v_i) = \Pi_q,\ g(v_j) = \Pi_r,$ and $(v_i, v_j) \in E \Leftrightarrow h((v_i, v_j)) = (\Pi_q, \Pi_r)$.

We can conclude that for $i, j = 1, ..., n$, and for $q, r = 1, ..., u$, $\forall (\Pi_r, \Pi_s) \in \Phi,\ \exists v_i, v_j \in V$ such as $g(v_i) = \Pi_r,\ g(v_j) = \Pi_q, h((v_i, v_j)) = (\Pi_r, \Pi_q)$.

Figure 2 illustrates the partition algorithm with this notion. The set M represents the main memory, where $M = \{(q, j),\ q \in \{1...u\},\ j \in \{1...k\}\}$. Note that a sampling-set Π_q is assigned to a partition S_j through its index.

When an element t of the unbound stream T arrives, we obtain its vertex. If we have assigned the set which that vertex belongs to, its index will belong to M. So, the partitioner has to send the vertex and its edges, t, to the already assigned partition S_i ($S_i = S_i \cup \{t\}$). On the contrary, if it is the first

Input: Unbound stream T
  $M = \emptyset$
  for all $t \in T$ **do**
    let v = get vertex $v \in V$ from t
    let $q = g(v)$
    if $\exists i \in \{1..k\} \mid (q, i) \in M$ *where* $S_i \in P$ **then**
      Send t to i partition node
      In partition node i $S_i = S_i \cup \{t\}$
    else
      $i = PartitionHeuristic(t, P)$
      $M = M \cup \{(q, i)\}$
      Send t to i partition node
      In partition node i $S_i = S_i \cup \{t\}$
    end if
  end for

Figure 2. Vertex Partition Algorithm

time a vertex of that set arrives, the partitioner computes the optimal partition for it, using the partition heuristic. Then, it adds the corresponding summary to M.

The main advantage is that the partition heuristic only depends on $\frac{n}{l}$ in memory terms and the algorithm has to partition $\frac{n}{l}$ vertex, instead of n.

The analysis of partition heuristic is out of the scope of this paper. The only requirement is that it has to be computed in constant time. In the experimentation phase, we have selected Fennel [8].

In our scheme, local information in each partitioner is updated with a frequency f; so, some information might be incoherent in this interval. In Section V, we show the relationship among the sampling function, the update period and the approximate solution.

A. Sampling functions.

Functions g and h, which are defined in Section IV, are light functions ($g, h \in O(1)$), and the information used to assign one element to one set has to be known *a priori* for each partitioner. In other words, as we want an easy distribution of the algorithm, the decision has to be made without taken into account the previous elements. Sampling function cannot depend on the arriving order. We consider the number of elements in a set constant, so $|\Pi_1| = |\Pi_2| = ... = |\Pi_{|\Psi|}| = l$

We propose the following sampling functions, which are based on the vertex identifier.

- **Hash function.** Each vertex $v_i \in V$ goes to a set depending on its identification $i, j = 1..n$ as follows:

$$g(v) = i \ mod \ l$$
$$|\Psi| = \frac{n}{l}$$
$$h((v_i, v_j)) = (\Pi_{i \ mod \ l}, \Pi_{j \ mod \ l}) \quad (3)$$

- **Consecutive assignation.** If the identification of a node has implicit an order (e.g., it is a number), we can build sampling-sets sequentially. In some situations (e.g., BFS and DFS model), this sampling function has more sense, because as elements arrive in a determined

way, connected elements go to the same set and to the same partition.

$$\forall i,j = 1..n,\ g(v_i) = i\ div\ l$$
$$|\Psi| = \frac{n}{l}$$
$$h((v_i,v_j)) = (\Pi_{i\ div\ l}, \Pi_{j\ div\ l}) \quad (4)$$

V. ANALYSIS.

Once we have proposed a model, we are going to analyse how much memory it needs in the partition stage and how much messages are sent from partitions to partitioners. We do not take into account the normal stream traffic because it is implicit to the model.

A. Memory Bound

Theorem 1: Given a graph $G = (V,E)$, a sample size l and a single-pass partition algorithm ALG which produces λ' cutting edges with $O(n)$ memory bound; there exists an (ε,δ)-approximation of cutting-edge fraction λ, with a $O(\frac{n}{l})$ memory bound in each parallel partitioner and $\varepsilon \in O\left(\sqrt{\frac{3lkln(\frac{1}{\delta})}{m[(l-1)(k-1)+\lambda' k]}}\right)$.

Proof: Memory reduction is achieved sampling incoming vertices into sets, and the sets are used as input of the algorithm ALG. With a sampling size of l, the memory bound belongs to $O(\frac{n}{l})$. In order to calculate the accuracy of the solution, we define a set of random variables X_{ij}, where $X_{ij} = 1$ if $v_i \in S_p, v_j \in S_q$ and $q \neq p, i,j = 1..n, p,q = 1..k$. By the law of total probability,

$$P(X_{ij} = 1) = \frac{(l-1)(k-1)}{lk} + \frac{\lambda'}{l} = p \quad (5)$$

As $\lambda' < p$, by Chernoff bound:

$$P\left[\frac{\sum X_{ij}}{m} \geq (1+\varepsilon)\lambda'\right] \leq e^{-\frac{\varepsilon^2}{2+\varepsilon}mp} \quad (6)$$

As $\lambda' < 1$ and $p > 0$, then $\varepsilon < 1$, so $-\frac{\varepsilon^2}{2+\varepsilon} < -\frac{\varepsilon^2}{3}$:

$$\varepsilon \in O\left(\sqrt{\frac{3lkln(\frac{1}{\delta})}{m[(l-1)(k-1)+\lambda' k]}}\right)$$

■

B. Number of sent messages

Theorem 2: Given a graph $G = (V,E)$, s distributed partioners, a sampling size l, an update frequency f and a single-pass partition algorithm ALG which produces a λ' cutting edges percent with a $O(n)$ memory bound; there exists an (ε,δ)-approximation of λ with a $O(\frac{n}{l})$ memory bound in each distributed partitioner and a $O(\frac{ns}{\sigma f})$ global distributed complexity, where σ is the incoming elements per time unit and $\varepsilon \in O\left(\sqrt{\frac{3ln(\frac{1}{\delta})}{m\left[1-\left(e^{\frac{-l\sigma f(\sigma f-1)}{2n}}\frac{\lambda'}{l}\right)\right]}}\right)$.

Proof: It is easy to see that the number of sent messages from partitions to partitioners depends on the update period f and on the number of partitioners s. In a f period, the system sends s messages. If $sigma$ elements arrive in one time unit, the entire graph arrives in $\frac{n}{\sigma}$. Then, in $\frac{n}{\sigma}$ periods, it sends $\frac{ns}{\sigma f}$.

Now let us calculate the accuracy of the solution. We consider that ALG will be better than a random partition algorithm ($\lambda' \in O\left(\frac{k-1}{k}\right)$). Therefore, the worst case happens when a vertex v_i whose $g(v_i)$ has been assigned in the same period f is assigned again by the random partition algorithm. The probability of get σf unique elements from n/l groups is (a.k.a. birthday problem):

$\frac{\frac{n}{l}-1}{\frac{n}{l}} \times \frac{\frac{n}{l}-2}{\frac{n}{l}} \times ... \times \frac{\frac{n}{l}-(\sigma f-1)}{\frac{n}{l}}$ and by a Taylor expansion its upper bound is $e^{\frac{-l\sigma f(\sigma f-1)}{2n}}$.

In order to simplify our calculus, we consider that the probability of generating a cutting edge by a random partition algorithm is about 1 (reasonable assumption for big k), so:

$$P(X_{ij} = 1) \approx 1 - \left(e^{\frac{-l\sigma f(\sigma f-1)}{2n}}\frac{\lambda'}{l}\right) = p \quad (7)$$

And by Chernoff bound:

$$P\left[\frac{\sum X_{ij}}{m} \geq (1+\varepsilon)\lambda'\right] \leq e^{-\frac{\varepsilon^2}{2+\varepsilon}mp} \quad (8)$$

As $\lambda < 1$ and $p > 0$, then $\varepsilon < 1$, so $-\frac{\varepsilon^2}{2+\varepsilon} < -\frac{\varepsilon^2}{3}$:

$$\varepsilon \in O\left(\sqrt{\frac{3ln(\frac{1}{\delta})}{m\left[1-\left(e^{\frac{-l\sigma f(\sigma f-1)}{2n}}\frac{\lambda'}{l}\right)\right]}}\right)$$

■

The number of sent messages per time unit is $\frac{s}{f}$. If we consider a distributed architecture with a distributed memory, the traffic between the memory and the partitioners per time unit is 2σ. For big values of σ, the bound of our system is better than the distributed memory approach.

VI. EVALUATION

We have implemented our model on a real environment in order to test it. Among the available open source distributed Data Stream Management Systems that are available in a distributed and open source version, we have chosen Storm [18], due to its flexibility to deploy the distributed infrastructure over the machines. The implementation has been developed using Java. We have set a Storm cluster of eight workers and one master. Each machine has one core and 2 GB of memory size.

In addition, we have executed the PageRank mining algorithm over the obtained partition in a GraphLab cluster [19]. GraphLab is a distributed graph processing engine which provides several data mining algorithms. Moreover, PageRank [4] is a graph mining algorithm which is used to rank elements in a network.

 ISBN: 978-1-61208-415-2

A. Datasets.

The datasets we have used to test our system are in Table I. *PL*, *WS* and *BA* datasets are synthetic, created by the Networkx package. We have used these datasets because Web Network and social graphs can be modelled by a power law graph, [20]. WS is a Watts-Strogatzsgraph model [21] and BA is a Barabasi-Albert graph [22]. *Amazon**, *Wiki-talk* and *LiveJournal1* are real datasets. The *amazon** dataset represents co-purchasing information of Amazon. If a product i is co-purchased with a product j, there is an edge from i to j in the graph. The information was collected in March 12 2003, May 05 2003 and June 01 2003. In *Wiki-talk* dataset, each vertex represents a user, and an edge from i to j represents that the user i has edited, at least once, the talk page of user j. The information was recollected in January 2 2008. In *LiveJournal1* dataset, each link between vertices (users) represents a friendship relation.

TABLE I. LIST OF USED DATASETS

Dataset	Vertices	Edges
WS10000	10000	134944
WS100000	100000	3997464
BA10000	10000	134841
BA100000	100000	3548775
PL10000	10000	134766
PL100000	100000	4047486
amazon0312	400727	2349869
amazon0505	400727	2439437
amazon0601	400727	2443311
LiveJournal1	4843953	42845684
Wiki-talk	2388953	4656682

B. Experimentation and evaluation.

We use λ and ρ metrics (see equations (1), (2)) as measures of the quality of the obtained partition.

In experiments, we have used Fennel [8] as best partition heuristic. We have measured the relationship among the quality of the partition (through λ and ρ), the sampling size l and the number of partitioners s. In addition, we have measured the amount of used memory and the impact of these parameters in the execution of a mining algorithm (PageRank).

1) Partition quality: In Figure 3, we observe how sample size l affects λ value. We have partitioned *amazon0312* dataset into 32 partitions. The experiment has been made with different incoming orders (Random and BFS) and Hash and Consecutive sampling functions (see equations (3), (4)). The first measure, $l = 1$, is equivalent to the one obtained in Fennel partition algorithm, and the last one corresponds to the situation when there is only one set per partition, $l = \frac{n}{k}$, which is equivalent to a random partition strategy. With two elements per group, the number of cutting edges increases significantly compared to Fennel, but it is better than the Random partitioner. In our results, the kind of sampling function used affects the quality of the partition. In a BFS incoming ordering, the results are better for a consecutive assignment function. This kind of order is naturally obtained in social and web graphs because they are obtained by crawlers.

2) Used Memory: We can see that the maximum used memory depends on l. In Table II, the results for the *LiveJournal* dataset are shown. In this case, we have partitioned into eight partitions with a BFS order and a consecutive assignment function.

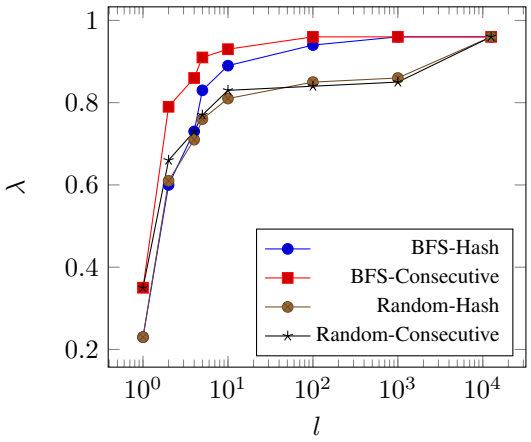


Figure 3. λ versus l in *amazon0312* dataset for different incoming orders and sampling functions.

TABLE II. VARIATION OF λ AND ρ IN LIVEJOURNAL1 DATASET WITH EIGHT PARTITIONS

l	λ	ρ	Used Memory (MB)
1	0.5	1.01	245.99
2	0.56	1.01	135.63
10	0.68	1.01	27.60
605495	0.96	1	0

In Figure 4, we can see the RAM memory required to store the sampling sets. As it is natural, when the number of elements per group increases, the total memory decreases. Note that with $l = 1$, the total used memory is 20,8 MB. Approximately, this is 0.052 kB per element, so we cannot process a web network graph (50 billion vertices [3]) with the Fennel algorithm.

TABLE III. VARIATION OF λ AND ρ IN *amazon0312* DATASET WITH 6 PARTITIONERS AND 32 PARTITIONS

l	λ	ρ
2	0.8	1.23
5	0.81	1.1
10	0.83	1.12
100	0.85	1.19
1000	0.91	1.18
12532	0.96	1

3) Distributed Partitioners: The number of partitioners s affects the quality of the partition, because they manage local information. In Table III, we show the results for $s = 6$. We have used contiguous grouping strategy and BFS arrival ordering. Experimental results show that with bigger sets, the performance is similar to the single loader. This is because with bigger sets, the number of partitioned elements is small, so the balanced load factor decreases.

4) PageRank Processing: The last experiment we have made is to execute a graph algorithm over obtained partition in GraphLab. We have used the *LiveJournal1* dataset and the PageRank algorithm for testing our system in a real scenario. We have chosen PageRank because is a well-known analytic over a graph, and we can use it to compare ourselves with

 ISBN: 978-1-61208-415-2

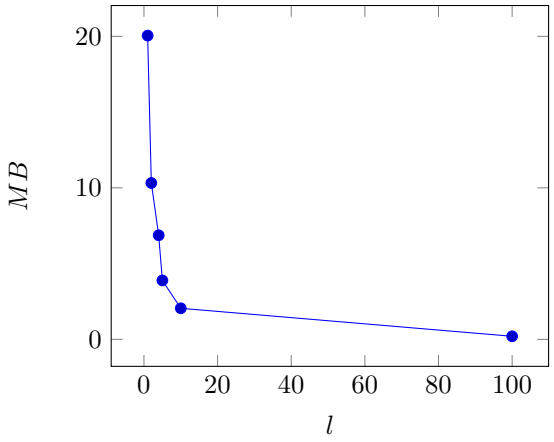


Figure 4. Required RAM memory to process *amazon0312* dataset versus sample size l

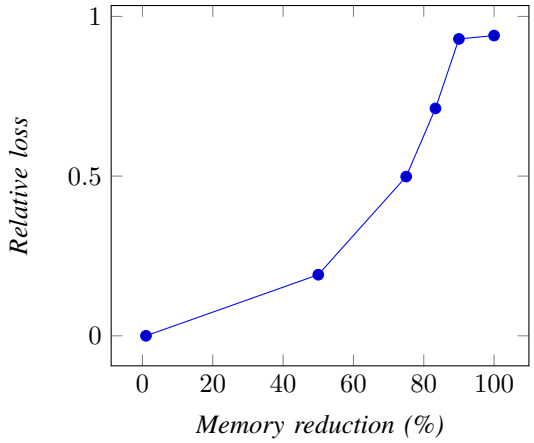


Figure 5. Relative loss of execution time of PageRank versus memory reduction.

previous works. The aim of the experiment is to measure the real trade-off between memory usage in partition stage and execution time of algorithm over the obtained partitions.

Figure 5 shows the loss in performance terms versus the memory reduction percent. We have calculated the loss using as reference the time the PageRank algorithm takes in a partition solution obtained by Fennel (equivalent to $l = 1$). The memory reduction has been calculated in the same way, and its relationship with l is straightforward. Last point refers to a Hash partition strategy, which does not use any memory, but almost doubles the execution time of PageRank. With a 50% memory reduction (equivalent to $l = 2$), the PageRank execution is only increased about 25%. For high values of l, we achieve a high memory reduction ($l = 10$ equals 90% of memory reduction), however the execution time of an analytic is similar to the obtained with a Hash partitioner.

VII. Conclusion and Future Work

In our work, we have proposed a scalable model which allows to partition large scale streaming graphs in an efficient way. To reduce memory usage, we have sampled vertex of incoming graph to compose a subgraph. We have used this subgraph to partition the original graph, with a single-pass generic algorithm. The information consistency is maintained updating local state in each partitioner with information from the partitions. In addition, we have calculated the memory bound, the introduced error and the distributed complexity of the model.

Our solution proposes a trade-off between available memory and processing time. With our sampling functions, not having a global knowledge of the graph does not cause a significant loss in performance terms. In our experiments, we show that a 50% reduction in RAM memory only increases the processing time of the PageRank algorithm a twenty five percent.

One future investigation line is to adopt the sampling model to more complex scenarios, like weighted or evolving graphs.

Acknowledgement

This work was partially supported by the Spanish Ministry of Economy under the program "Programa de I+D+i Estatal de Investigación, Desarrollo e innovación Orientada a los Retos de la Sociedad", project identifier TIN2013-40809-R, and by COSMOS, research group recognized by the Aragonese Government. V.Medel was the recipent of a fellowship from Departamento de Industria e Innovación of the Diputación General de Aragón.

References

[1] S. Muthukrishnan, Data streams: Algorithms and applications. Now Publishers Inc, 2005.

[2] "Yahoo! AltaVista Web Page Hyperlink Connectivity Graph 2002," 2015, URL: http://webscope.sandbox.yahoo.com/catalog.php [retrieved: May, 2015].

[3] G. Malewicz et al., "Pregel: a system for large-scale graph processing," in Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. ACM, 2010, pp. 135–146.

[4] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.

[5] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," Physical review E, vol. 69, no. 2, 2004, p. 026113.

[6] T. Schank and D. Wagner, "Finding, counting and listing all triangles in large graphs, an experimental study," in Experimental and Efficient Algorithms. Springer, 2005, pp. 606–609.

[7] R. Pagh and C. E. Tsourakakis, "Colorful triangle counting and a mapreduce implementation," Information Processing Letters, vol. 112, no. 7, 2012, pp. 277–281.

[8] C. Tsourakakis, C. Gkantsidis, B. Radunovic, and M. Vojnovic, "Fennel: Streaming graph partitioning for massive scale graphs," in Proceedings of the 7th ACM international conference on Web search and data mining. ACM, 2014, pp. 333–342.

[9] I. Stanton and G. Kliot, "Streaming graph partitioning for large distributed graphs," in Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2012, pp. 1222–1230.

[10] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and issues in data stream systems," in Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. ACM, 2002, pp. 1–16.

 ISBN: 978-1-61208-415-2

[11] T. Feder, P. Hell, S. Klein, and R. Motwani, “Complexity of graph partition problems,” in Proceedings of the thirty-first annual ACM symposium on Theory of computing. ACM, 1999, pp. 464–472.

[12] Z. Bar-Yossef, R. Kumar, and D. Sivakumar, “Reductions in streaming algorithms, with an application to counting triangles in graphs,” in Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, 2002, pp. 623–632.

[13] J. Feigenbaum, S. Kannan, A. McGregor, S. Suri, and J. Zhang, “Graph distances in the streaming model: the value of space,” in Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, 2005, pp. 745–754.

[14] D. Garcıa-Soriano and K. Kutzkov, “Triangle counting in streamed graphs via small vertex covers,” Tc, vol. 2, 2014, p. 3.

[15] J. M. Ruhl, “Efficient algorithms for new computational models,” Ph.D. dissertation, Citeseer, 2003.

[16] A. D. Sarma, R. J. Lipton, and D. Nanongkai, “Best-order streaming model,” in Theory and Applications of Models of Computation. Springer, 2009, pp. 178–191.

[17] D. LaSalle and G. Karypis, “Multi-threaded graph partitioning,” in Parallel & Distributed Processing (IPDPS), 2013 IEEE 27th International Symposium on. IEEE, 2013, pp. 225–236.

[18] “Apache Storm Website,” 2015, URL: http://storm.apache.org [retrieved: May, 2015].

[19] Y. Low, J. Gonzalez, A. Kyrola, D. Bickson, C. Guestrin, and J. M. Hellerstein, “Graphlab: A new framework for parallel machine learning,” arXiv preprint arXiv:1006.4990, 2010.

[20] M. Faloutsos, P. Faloutsos, and C. Faloutsos, “On power-law relationships of the internet topology,” in ACM SIGCOMM Computer Communication Review, vol. 29, no. 4. ACM, 1999, pp. 251–262.

[21] D. J. Watts and S. H. Strogatz, “Collective dynamics of small-world networks,” nature, vol. 393, no. 6684, 1998, pp. 440–442.

[22] R. Albert and A.-L. Barabási, “Statistical mechanics of complex networks,” Reviews of modern physics, vol. 74, no. 1, 2002, p. 47.

 ISBN: 978-1-61208-415-2

Sketch of Big Data Real-Time Analytics Model

Bakhtiar M. Amen
School of Computing and Engineering
The University of Huddersfield
Huddersfield, UK
e-mail: bakhtiar.amen@hud.ac.uk

Joan Lu
School of Computing and Engineering
The University of Huddersfield
Huddersfield, UK
e-mail: j.lu@hud.ac.uk

***Abstract*— Big Data has drawn huge attention from researchers in information sciences, decision makers in governments and enterprises. However, there is a lot of potential and highly useful value hidden in the huge volume of data. Data is the new oil, but unlike oil data can be refined further to create even more value. Therefore, a new scientific paradigm is born as data-intensive scientific discovery, also known as Big Data. The growth volume of real-time data requires new techniques and technologies to discover insight value. In this paper we introduce the Big Data real-time analytics model as a new technique. We discuss and compare several Big Data technologies for real-time processing along with various challenges and issues in adapting Big Data. Real-time Big Data analysis based on cloud computing approach is our future research direction.**

Keywords - Big Data Analytics; Real-time Analytics; Big Data state-of-the-art

I. INTRODUCTION

The term "Big Data" is universal and has gained popularity within the domain of scientist, bioinformatics, geophysics, astronomy and meteorology [1]. In fact, all Big Data has blind spot areas in which data are missing, scarce, or otherwise unrepresentative of the data domain [2]. Big Data analytics enable enterprise and scientists to extract usable information out of enormous, complex, interconnected and varied datasets. However, from 2.8 Zettabytes of global data only 0.5% of these data was analyzed in 2012 [3]. In addition to this, current Big Data techniques and technologies are incapable of storing, processing or analyzing data, as data is not extracted by particular scientific disciplines (e.g., bioinformatics, geophysics, astronomy and meteorology). The way in which people think about data and data analysis will gradually change as well, in addition to the technological possibilities. Thanks to the latest internet technologies, the potential for harnessing all that can be measured and analysed using solid data, intelligent sensors and filtering has never been as promising and lucrative as today, at the dawn of the digital era [4].

The Big Data paradigm consists of batch and real-time processing [5]. The batch process focuses entirely on structured and semi-structured data. Likewise, the goal of real-time processing paradigm is to deal with velocity of Big Data such as processing streaming data but with low latency. This paper aims to review the background of Big Data and compare several Big Data real-time processing technologies, as well as introduce the new real-time Big Data analytics model.

The rest of this paper is organized as follows: In Section 2 we briefly overview some concepts of Big Data including its definition, characteristics and size. Section 3 presents the Big Data domains. Section 4 presents related work. Section 5 presents an evaluation and discussion. The article culminates in a conclusion and recommendation for future work.

II. BACKGROUND

In this section, we present Big Data definitions, its characteristics, followed by the Big Data revenue and the size of global data. Next, we present a Big Data technology map.

A. Big Data Definition

Big Data is one of the key buzzwords in the current technological landscape, but there is no agreed definition by either academia or industry. Chen et al. [6] defined Big Data as "Datasets which could not be captured, managed, and processed by general computers within an acceptable scope". Hashem et al. [7] also defined Big Data as "a term utilized to refer to the increase in the Volume of data that is difficult to store, process, and analyze through traditional database technologies". However, these definitions basically state the most obvious dimensions of Big Data Volume, Variety, Velocity and Veracity. Whereas, the data flows in today's digital era are being produced around the clock and all over the world.

B. Big Data Characteristics

The conjunction of these four dimensions helps both to define and distinguish Big Data. Volume refers to the amount of data from Terabyte to Petabyte and Exabyte to Zettabyte [6]. Variety refers to various data sources collected from web logs, social networks, machines, sensors, transactions and the internet of things, in different formats of semi-structured and unstructured [8]. Velocity refers to the speed at which data is generated and the speed of data transfer [7]. Data has become an extremely valuable factor in business productivity and the opportunity to discover new value from it. The 4V's of Big Data are shown in Figure 1.

 ISBN: 978-1-61208-415-2

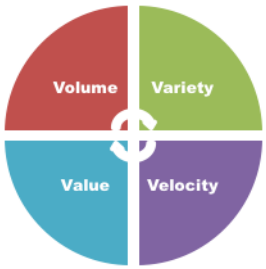


Figure 1. The characteristics of Big Data

C. Size of Global Data

The size of digital data has been growing at an increasing rate. Figure 2 depicts the size of created data volumes in percentages across the USA, West Europe, India and the rest of the world. According to the International Data Corporation (IDC) study, the size of global data in 2009 was 1.8 Zettabyte, it increased to 8 Zettabyte in 2015 [9]. It is doubling in size every two years, and by 2020 the digital universe data is estimated to be 44 Zettabytes [10].

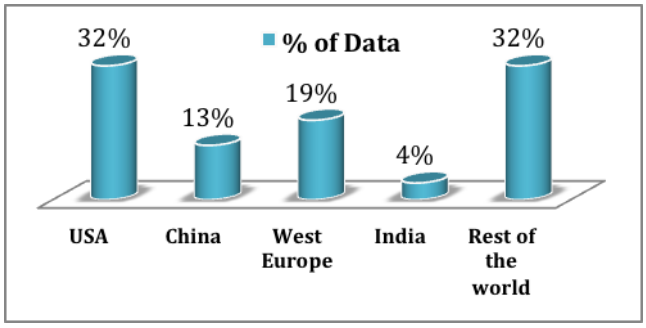


Figure 2. The scale of global data generates in percentages.

Furthermore, the explosive growth of global data increased rapidly. In fact, 90% of the world's entire data was created since 2012 [11], whereas only 10% of all these data is structured data compare to 80% is unstructured data [11]. Figure 3 depicts the scale of global structured data versus unstructured data.

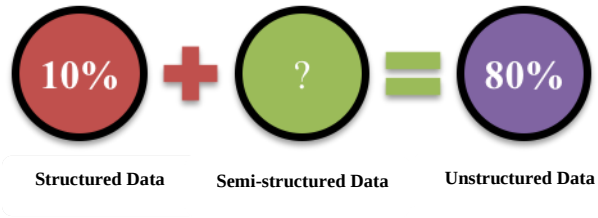


Figure 3. The scale of universe data format

D. Big Data Revenue

Manyika et al. [12] estimated that the power of Big Data analytics guaranteed 60% of potential revenue through new opportunities from location-aware and location-based services. In reality, the ambiguous demand in the Big Data era is more related to business insights since the 4Vth Value of Big Data has been introduced. Big Data revenue increased from $3.2 billion to $16.9 billion between 2010 and 2015 [6]. However, the potential value to consumers, business and users are estimated to be $700 billion in the next ten years [7].

E. Big Data Real-time State- of-the-art

Hadoop is known as innovative in Big Data analytics, since Hadoop has the ability to touch 50% of the global data by 2015 [13]. In fact, Hadoop and MapReduce have been criticized by both academia and enterprises for their real-time limitations. The MapReduce programming model is an open-source version of Hadoop [13][14]. Fan et al. [14] stated that Hadoop made a world record in sorting one petabyte of data within 16.25 hours and one terabyte of data in only 62 seconds. Furthermore, the Hadoop ecosystem consists of several projects as introduced only the real-time applications in Figure 7.

Twitter has developed storm in 2011 [10][15] for data streaming processing. Storm is an open source and it has been improved in scalability while maintaining a low latency for real-time data stream processing, which integrates with other queuing and bandwidth systems. Storm consists of several moving parts, including the coordinator (ZooKeeper), state manager (Nimbus) and processing nodes (Supervisor). Yahoo has developed S4 in 2010 [13][14][16] for data stream parallel distributing processing. Kafka also developed LinkedIn in 2011 [16] for the purposes of messaging processing. Spark [9] Stream is an extension of Spark that supports continuous stream processing. In practice, some other new computing models have recently been introduced for stream data processing (e.g., GraphLab and Dryad), which are suitable for machine learning and data mining programming models [16] .

III. BIG DATA DOMAINS

In this section, we describe some of the challenges and issues of Big Data in several disciplines from both industry and scientific perspectives.

A. Big Data in the Bio-Medical Sector

Kambatla et al. [17] highlighted that “healthcare and human welfare is one of the most convincing applications of Big Data analytics, it is a fastest growing datasets”. In fact, a large amount of medical data sources like RMI scans, bioscience data, and genomic data are becoming more complex and difficult to be captured, storage, and analyzed [18]. Although, China attempts to collect and store 30 million of traceable biological samples by 2015 [19]. Manyika et al. [12] stated that every year the USA has wasted more than $2 trillion in healthcare sectors. Implementing Big Data analytics technique has helped the US to save $300 billion as well as helping Europe to save over $149 billion. In addition, bio-informatics requires new advanced computational techniques to support efficient knowledge discovery.

 ISBN: 978-1-61208-415-2

B. Big Data in Enterprise

Facebook, Google, Yahoo and Falcon are creating large scale of data. As an example, Wal-Mart produced over 1 million customer transactions per hour across 6000 stores [6]. Amazon Web Services (AWS) has also been successful in IaaS services with 70% of their market share including the most popular Elastic Compute Cloud (EC2). A Simple Storage Service (S3) enables the processing of 500,000 queries over millions of terminal operations from third party sellers each day [6][20]. Akamai also managed to analyze 75 million events per day. However, the most observable domain in Big Data analytics is value [20]. Hence, Data has become extremely valuable in enterprise to produce productivity and business predictions.

C. Big Data in Scientific Research

Every day NASA solar observatory and telescopes are capturing more than 1.6 TB of high quality images and collecting 140 TB data from the large synoptic survey telescope [21]. Likewise, one space satellite generates over 800 gigabytes of data on a daily bases. In 2012, the Earth Observing System Data and Information System (EOSDIS) also succeeded in distributing more than 4.5 million gigabytes of data per day [20]. However, physicists and astronomists have made numerous efforts to engage with massive crowded data for many years to test the novelty of our universe.

D. Big Data in Engineering

The key challenge in the area of engineering is the discovery of techniques that are able to process machinery and the internet of things data. These sources are creating massive amounts of data through embedded networking and real-time approaches. The size of the internet of things data is estimated to be one trillion by 2030 [20]; this includes 350 billion annual meter readings. The volume of data generated in engineering is by a wide range of sensors, through power plants, machinery data and GPS as well as electronic devices [22].

IV. RELATED WORK

Patel [23] highlighted several issues and challenges in storing, processing and analyzing data in real-time. The author argued that highly efficient algorithm and technology will enhance the accuracy of valuable information [24]. Ranjan [24] investigated in different Big Data applications and discussed their differences from traditional analytics, and he also described the new solutions for real-time Big Data analytics. Kambatla et al. [17] implemented several projects in a real-time data caching and processing graph in one of Google's distributing systems. The author also highlighted that current technology such as Hadoop incapable of processing large-scale of graphs. Hadoop mainly consists of two components; Hadoop File System (HDFS) and Programming model (Map Reduce) [25]. HDFS stores huge data set reliably and streams it to user application at high bandwidth and MapReduce is a framework that is used for processing massive data sets in a distributed fashion over a several machines. It has two parts- job tracker and task tracker [26].

Hu [27] proposed HACE theorem which characterizes the features of the Big Data revolution, and recommends the Big Data processing model, from a data mining perspective. Moreover, the rapid growth of complex diversity and dimensionality of the Remote Sensor (RS) lies in collected metadata to analyze as stated in [28]. The recent lower level parallel programming was comprehensively engaged with RS image processing along with a multi-level hierarchical cluster. Hence, parallel programming is required for RS applications to predict accurate results. According to Ma et al. [28], the current Big Data analytics model is beyond the capabilities of processing and analyzing real-time Satellite Data.

The scale of remote satellite data is depicted in Figure 4; this scale demonstrates the volumes of satellite Data per day as well as per year across the world.

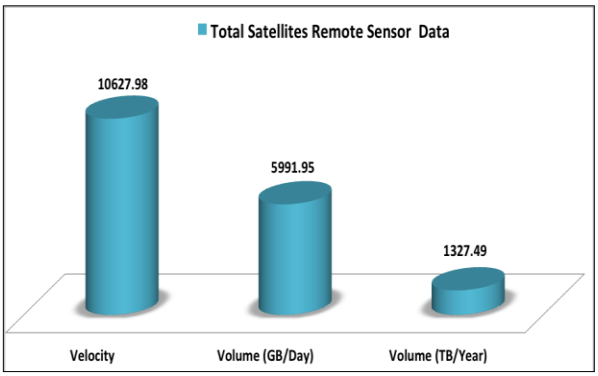


Figure 5. Scale of Global Satellite Remote Sensor Data.

Two Big Data real-time/stream analytics model were found in our literature, known as Simith's Big Data real-time analytics Model [29] and Big Data life cycle management model [30]. Khan et al. [16] proposed Big Data life cycle management model using the technologies and terminologies of Big Data. The author's proposed data life cycle consists of: Data Acquisition/Generation, Data Collection, Data storing (temporarily/permanently), and Data Analysis. Likewise, Barlow [29] presented Smith's five phases of the real-time Big Data analytics model which includes: Data extraction, development model, validation and deployment, real-time scoring, and model refresh.

Barlow also stated that the correct analytics model is necessary to process heterogeneous data in real time. Furthermore, this model is utilized from the high-performance of data mining, predictive analytics, text mining, and data optimization to enhance the decision makers [1][31]. In fact, the heart of any prediction system is the Model, for instance, a credit card fraud prediction system could leverage a model built using previous credit card transaction data over a period of time.

 ISBN: 978-1-61208-415-2

TABLE I. SMITH'S BIG DATA ANALYTIC MODEL

Analytic Model	Descriptions
Data Extraction/Distillation	Like unrefined oil, heterogeneous data types are messy and complex. Emerging new extracting models and performing accurate analysis are necessary and challenging to handle unstructured data [11][18].
Development Model	In this phase, the model process consists of speed, flexibility productivity, and reproducibility.
Validation and Deployment	Extracting fresh data and running against the model and comparing the results with other existing models leading into productivity [13].
Real-time Scoring	Data in real-time scoring is triggered by actions at the decision layer. At this phase of the process, the deployed scoring rules are "divorced" from the data in the data layer or data mart [21].
Model Refresh	Data is always changing, it is necessary to refresh the data and refresh the model built on the original data. Simple exploratory data analysis is also recommended.

Hu et al. [1] categorized Big Data analytics into Descriptive, Predictive and Prescriptive. Descriptive analytics focuses on historical data and the description of what occurred previously from Data visualization results. Predictive analytics focuses on future probabilities and describes the business value outcome. Advanced analytics [31] is known as Prescriptive analytics which address the decision making efficiently. For example, simulation is used to analyze complex systems to gain insight into system behavior and identify issues and optimization techniques are used to find optimal solutions under given constraints. However, only about 3% of companies are utilizing prescriptive analytics to predict future events according to a recent Gardner Research survey [32].

V. EVALUATION AND DISCUSSION

Despite the availability of new technologies for handling massive amounts of data at incredible speeds, the real promise of advanced data analytics lies beyond the area of pure technology. The existing Big Data analytics appears to be suffering from a lack of effectiveness compared to the speed of real-time data volume. Thus, Big Data real-time analytics has been proposed to describe the advanced analysis methods or mechanisms for massive data [11]. In fact, increasing the heterogeneous data in the real-time monition from various data sources (e.g., The Internet of Things, multimedia, social networking) plays a significant role in Big Data. In addition to these, the new real-time model shown in Figure 6 is required. Banerjee [33] highlighted the traditional analytics versus real-time analytics in Figure 5. Banerjee also compared several parameters in each feature from storage cost to support cost. It shows that Big Data analytics is more reliable in terms of data speed, time and velocity compared to traditional analytics. This highlights the key differences between the realities of yesterday's analytics and the predictions for today's Big Data analytics.

	Traditional Analytics	Big Data Analytics
Storage Cost	High	Low
Analytics	Offline	Real-time
Utilizing Hadoop	No	Yes
Data Loading Speed	Low	High
Data Loading Time	Long	Average 50%-60% faster
Data Discovery	Minimal	Critical
Data Variety	Structured	Unstructured
Volume	Gigabyte, terabyte	Petabyte, Exabyte, Zettabyte
Velocity	Batch	Real-time
Administration Time	Long	Average 60% faster
Complex Query Response Time	Hours/days	Minutes
Data Compression Technique	Not matured	Average 40%-60% more data compression
Support Cos	High	Low

Figure 7. The Traditional Analytics versus Big Data Analytics [33].

In addition to these, we implemented the new real-time analytics model from Smith's model as depicted in Figure 6, because Smith's model was precisely based on data mining and text mining [28]. As shown in Figure 6, this model consists of five phases: Data Extraction, Data Cleaning/Filtering, Data Analysis, Data Visualization, and Decision-making.

In the Data extraction phase, Data is required to be processed by one of the real-time technologies such as Storm and S4 as highlighted in Figure 7. Data must be cleaned before being transformed for analyzing to unlock the hidden potential value from it. Therefore, the second phase of filtering technique is required for two reasons. Firstly, data intent to lies in the extracting stage as indicated in [28]. Secondly, processing data without filtering means invalid results. Data visualization has to communicate and predict data through graphics to aid decision-making through sophisticated analytics results. In addition to this, advanced analytics for massive data is required as a new solution to effectively improving decision making in the final phase. As a result, the process of Real-time data analytics is still a challenging task and the model requires an advanced computational and robust real-time algorithm to predict it efficiently.

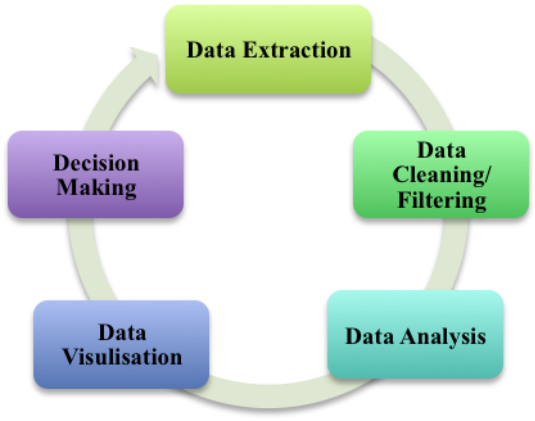


Figure 8. The Big Data Real-time analytics model.

Furthermore, selecting an appropriate real-time analytics model and technology depends on data objects. As depicted in Figure 7, we highlighted the Big Data real-time processing state-of-the-art in terms of its developers, programming model, capabilities, and limitations of data structure types. We highlighted each application's advantages and disadvantages as well as their architectures. The results show that, first, real-time processing is becoming more important in real-time analytics, likewise batch processing remains the most common data processing paradigm [28, 34].Second, most of the systems adopted a graph programming model, because the graph processing model can express more complex tasks. Third, all the systems support concurrent execution to accelerate the processing speed. Fourth, data stream processing models use memory as the data column storage to achieve higher access and processing rates, whereas batch-processing models employ a file system or disk to store massive data and support multiple visiting. Fifth, some of these real-time technologies were backed by partially fault tolerant and have limitations in their node backup as highlighted in Storm and S4 [1][13][14][15][16].

(e.g., semi-structured or unstructured), and this helped us to highlight their advantage and disadvantages.

In the second part of our study, we discussed Smith's five phases of the Big Data real-time analytics model as depicted in Table 1. Furthermore, we introduced our real-time Big Data analytics model as shown in Figure 6. Throughout our investigation, the real-time analytics appears to play a key role in Big Data and enrich the potential revenue. Likewise, it needs further research and collaborations between the scientists and industries to improve the real-time analytics bottleneck. In fact, a different storage mechanism is required, because all of the data cannot fit in a single type of storage area [35]. Hence, Cloud computing is playing an important role as it gives organizations the ability to store and analyze revolutionized data economically and offers extensive computing resources [16].

As result, the motivation for undertaken this research was an attempt to develop a real-time Big Data analytics framework which enable to enrich the decision-makers in the real-time monition. This research allowed us to identify the weaknesses of existing systems, and to design a roadmap of contributions to the state of the art.

	Big Data Streaming Analysis State-Of-The-Art							
	Developer	**Application**	**Programming Model**	**Specified Use**	**Structure Type**	**Advantages**	**Disadvantages**	**Architecture**
Storm - 2011 [3] [18] [28]	**Twitter**	Kafka, HBase (Storm-HBase) Twitter [23]	Directed Acyclic graph [3] [18]	Distribute real-time computation system for processing fast, large streams of data [18]	Un-structured, Real time Streaming process [18] [28][27]	Embeddable networking library API [1][3] Scalable, fault-tolerant, and is easy to set up and operate [3][18]	Partial fault tolerance [3] Lack of dedicated backup nodes [10]	Parallel-Distributed [23] Master-workers [3]
S4 – 2010 [3] [18] [25] [28]	**Yahoo**	Distributed Streaming process [3][18]	Directed Acyclic graph [3][18] [30]	Worker processes and execution, [3] Graph of Processing Elements [27]	Un-structured Real time streaming processing [3][18] [21] [27]	Distributed, Scalable, Fault-tolerant [21] [25] Pluggable platform [18] Easy develop applications [21]	Node fail data lose, Partial fault tolerance[3] [25] Lack of dedicated backup nodes [10]	Decentralised and systematic [23]
Kafka - 2011 [2][18][28]	**LinkedIn**	Messaging system Tool [28]	Distributed Messaging system [28]	Distributed, partitioned and replicated commit log services tools [18]	Real-time streaming process [3] Un-structured [18]	High-throughput stream of unchallengeable activity data [18][28]		Column store [22]

Figure 9. Big Data stream processing state-of-the-art

VI. CONCLUSION AND FUTURE WORK

In this research, we investigated on two research areas: in the first part of study, we presented the concepts of Big Data as well as some of the challenges and issues in both industry and scientific domains, followed by a comparison of several Big Data real-time processing technologies in terms of their capabilities and limitations as shown in Figure 7. However, each of these technologies were compared in terms of their architecture, programming model, data structure capabilities

Our future plan is to investigate on real-time analytics based on cloud computing and attempts to answer the following questions:

- Investigate on existing cloud paradigms and highlight their limitations in real-time analytics aspects.
- Develop an algorithm for the real-time analytics based on cloud computing.
- Determine how to test, implement and compare our results with other existing cloud computing technologies.

 ISBN: 978-1-61208-415-2

REFERENCES

[1] Y. W. Han Hu, Tat-Seng Chua, Xuelong Li, "Toward Scalable Systems for Big Data Analytics: A Technology Tutorial," vol. 2, pp. 652-687, 2014.

[2] J. J. Berman, Principles of big data: preparing, sharing, and analyzing complex information: Newnes, 2013.

[3] N. Khan, I. Yaqoob, I. A. T. Hashem, Z. Inayat, W. K. M. Ali, M. Alam, M. Shiraz, and A. Gani, "Big data: survey, technologies, opportunities, and challenges," TheScientificWorldJournal, vol. 2014, p. 712826, 2014.

[4] G. Aydin, I. R. Hallac, and B. Karakus, "Architecture and Implementation of a Scalable Sensor Data Storage and Analysis System Using Cloud Computing and Big Data Technologies," Journal of Sensors, vol. 2015, pp. 1-11, 2015.

[5] K. Kolomvatsos, C. Anagnostopoulos, and S. Hadjiefthymiades, "An Efficient Time Optimized Scheme for Progressive Analytics in Big Data," Big Data Research, 2015.

[6] M. Chen, S. Mao, and Y. Liu, "Big Data: A Survey," Mobile Networks and Applications, vol. 19, pp. 171-209, 2014.

[7] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. Ullah Khan, "The rise of "big data" on cloud computing: Review and open research issues," Information Systems, vol. 47, pp. 98-115, 2015.

[8] N. Khan, I. Yaqoob, I. A. T. Hashem, Z. Inayat, W. K. M. Ali, M. Alam, M. Shiraz, and A. Gani, "Big data: survey, technologies, opportunities, and challenges," vol. 2014, p. 712826, 2014.

[9] J. J. Berman, Principles of Big Data: Preparing, Sharing, and Analyzing Complex Information, 1st ed.: Morgan Kaufmann, 2013.

[10] D. R. John Gantz, "The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East," 2013.

[11] C. L. Philip Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on Big Data," Information Sciences, vol. 275, pp. 314-347, 2014.

[12] B. Brown, M. Chui, and J. Manyika, "Are you ready for the era of 'big data'," McKinsey Quarterly, vol. 4, pp. 24-35, 2011.

[13] C. Dobre and F. Xhafa, "Parallel Programming Paradigms and Frameworks in Big Data Era," International Journal of Parallel Programming, vol. 42, pp. 710-738, 2014.

[14] F. H. Jianqing Fan, and Han Liu, "Challenges of Big Data Analysis," National science review, vol. 1, pp. 293-314, Jun 2014.

[15] B. D. a. Zachary Miller a, William Deitrick a, Wei Hua, Alex Hai Wang, "Twitter spammer detection using data stream clustering," 2013.

[16] N. Khan, I. Yaqoob, I. A. Hashem, Z. Inayat, W. K. Ali, M. Alam, M. Shiraz, and A. Gani, "Big data: survey, technologies, opportunities, and challenges," TheScientificWorldJournal, vol. 2014, p. 712826, 2014.

[17] K. Kambatla, G. Kollias, V. Kumar, and A. Grama, "Trends in big data analytics," Journal of Parallel and Distributed Computing, vol. 74, pp. 2561-2573, 2014.

[18] M. C. James Manyika, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, Angela Hung Byers, "Big data: The next frontier for innovation, competition, and productivity," 2011.

[19] V. C. M. Leung, M. Chen, Y. Zhang, and S. Mao, Big Data: Related Technologies, Challenges and Future Prospects. DE: Springer Verlag, 2014.

[20] C. Esposito, M. Ficco, F. Palmieri, and A. Castiglione, "A knowledge-based platform for Big Data analytics based on publish/subscribe services and stream processing," Knowledge-Based Systems, vol. 79, pp. 3-17, 2014.

[21] J. Green, P. Schechter, C. Baltay, R. Bean, D. Bennett, R. Brown, C. Conselice, M. Donahue, X. Fan, and B. Gaudi, "Wide-field infrared survey telescope (WFIRST) final report," arXiv preprint arXiv:1208.4012, 2012.

[22] H. J. Watson, "Tutorial_ Big Data Analytics_ Concepts Technologies and Applica," Journals at AIS Electronic Library, vol. 34, pp. 1247-1268, 2014.

[23] A. B. Patel, M. Birla, and U. Nair, "Addressing big data problem using Hadoop and Map Reduce," 2012, pp. 1-5.

[24] B. W. Dan Vesset, Henry D. Morris, Richard L. Villars, Gard Little, Jean S. Bozman, Lucinda Borovick, Carl W. Olofson, Susan Feldman, Steve Conway, Matthew Eastwood, Natalya Yezhkova, "Market Analysis Worldwide Big Data Technology and Service " IDC Analyse Future, vol. 1, 2012.

[25] A. Katal, M. Wazid, and R. H. Goudar, "Big data: Issues, challenges, tools and Good practices," 2013, pp. 404-409.

[26] J. Nandimath, E. Banerjee, A. Patil, P. Kakade, and S. Vaidya, "Big data analysis using Apache Hadoop," pp. 700-703.

[27] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data Mining with Big Data," IEEE Transactions on Knowledge and Data Engineering, vol. 26, pp. 97-107, 2014.

[28] Y. Ma, H. Wu, L. Wang, and B. Huang, "Remote sensing big data computing: Challenges and opportunities," Future Generation Computer Systems, vol. in press, 2014.

[29] M. Barlow, "Real-Time Big Data Analytics: Emerging Architecture," pp. 24-25, 2013.

[30] R. Casado and M. Younas, "Emerging trends and technologies in big data processing," Concurrency and Computation: Practice and Experience, vol. 27, pp. 2078-2091, 2015.

[31] Vijay Srinivas Agneeswaran, "Big Data Analytics Beyond Hadoop " pp. 22-23, 2014.

[32] S. P. Soumya Sree Laxmi P. , "Impact of Big Data Analytics on Business Intelligence-Scope of Predictive Analytics," vol. Vol.5, 2015.

[33] A. Banerjee, "Big Data & Advanced Analytics in Telecom: A Multi-Billion-Dollar Revenue Opportunity" pp. 7-22, 2013.

[34] Y. Ma, H. Wu, L. Wang, B. Huang, R. Ranjan, A. Zomaya, and W. Jie, "Remote sensing big data computing: Challenges and opportunities," Future Generation Computer Systems, 2014.

[35] E. B. Dudin and Y. G. Smetanin, "A review of cloud computing," Scientific and Technical Information Processing, vol. 38, pp. 280-284, 2011.

 ISBN: 978-1-61208-415-2

Augmenting Data Files with Semantics for Coherency, Extensibility, and Reproducibility

John McCloud, Subhasish Mazumdar
Dept. of Computer Science
New Mexico Institute of Mining and Technology
Socorro, NM, U.S.A
email: {amccloud, mazumdar}@cs.nmt.edu

***Abstract*—Data files have traditionally been thought of as the input and output of programs, as well as their intermediaries. When the need for usage of data files by a diverse set of consumers was recognized, it was addressed primarily by the addition of metadata. This metadata is structured data, providing guidance regarding the use of the data. Unfortunately, this approach has proven inadequate for the myriad applications of today. We posed two questions of a very common and popular data file standard in bioinformatics. First, are the conclusions presented in such a file verifiable? Second, can one use the data to test for alternative conclusions? Our answers for both questions were negative. In this paper, we outline the problems we found and propose a remedy. While we have used bioinformatics as a case study, our results are more general.**

Keywords–Knowledge Representation; Comprehension; Semantics; Bioinformatics.

I. INTRODUCTION

How are data files designed? Traditionally, they have been designed to handle the needs of a computer program that consumes it, or as the output of a program for use by either humans or yet another program.

Thus, software systems exist that will take some particular data format and operate on it with satisfactory result. Such programs have canned understanding of the data and how it should provide some solution desired by a user. However, users themselves may neither be able to examine that data or make any sense of it at all. These individuals *must* rely on some software in order to express the high-level concepts that data tries to represent.

This problem exists because data usually lacks a kind of description of its content or guidance about its use. The solution to this problem has been *metadata*, but what amount of metadata is adequate? It is not clear where to draw the line; usually, metadata is constructed in an ad-hoc manner. We suggest that the answer be tied to an ontology, i.e., data files should be designed in terms of concepts defined in an ontology of the domain.

We consider bioinformatics data files in the very popular SAM file format and ask two questions: does the data format support (a) extensibility and (b) reproducibility? The first question asks whether or not researchers can use the included metadata to pose queries that pre-existing dedicated software (SAMTools) cannot answer.

Such extensibility is desired today as funding agencies are now insisting on the availability of created data sets for use by others; this has the promise of increasing research impact — perhaps exponentially. Reproducibility of results is a cornerstone of research. This too, we feel, must be supported by the published data sets.

In this paper, we report how we obtained negative answers to both questions of the SAM format. Further, we identify the gap as the lack of declarative functions or algorithms.

The issue of computational solutions in research has attracted renewed attention [1]. Currently, there is no accepted, standardized way for both code and data to be included alongside journal publications, and even the role a journal should play in vetting this additional information is not clear [2].

Research on data provenance is a promising line of attack: it attempts to bring lineage in the form of input, output, and so forth, presented as some workflow with clear beginning and end. myGrid, for example, creates such workflows from actionable steps in a process that can be saved and shared for re-use [3].

The Collaboratory for Multi-Scale Chemical Science (CMCS) is similar to myGRID, except that it has philosophical differences on what metadata means, and is able to present papers into related workflows. This is beneficial, since the scientific narrative and explanation of some methodology can be referenced and associated with previous, peer-reviewed research [4].

The solutions in myGrid and others, however, rely on middleware (such as Taverna [5]) to build and re-use the XML workflow constructions. A more light-weight approach is found in ESSW [6], which ties metadata and provenance to regular software by wrapping scripts around them. A script must be built for each actionable step (from input-to-output), and the script-writer is responsible for each piece of lineage information in the resulting provenance record.

myGrid and CMCS are both particularly interesting solutions, since they attempt to pair process workflow activities with domain-specific concepts. Tying such meaning to provenance records allows for researchers to understand, query, and make use of data more effectively.

The problem with all of these approaches is that they typically work on entire sets of data or files. What is needed is a solution that deals with sections or components of data *within* files. We imagine using the Resource Description Framework (RDF) [7] to annotate directly *within* data files themselves. RDF is a desirable choice for annotation, because its pairing with ontologies is understood [8], and RDF has been already been useful in provenance research, such as myGrid.

 ISBN: 978-1-61208-415-2

Furthermore, even with all this research, the entirety of *processes* that transform some initial data into some final result is not represented clearly in the data itself. While implementations of embeddable scripts into workflows are possible to view and maintain, internals of online web-services or programmed solutions can be opaque. The functions that are used in some computational solution may remain a black-box or exist in various languages that researchers of different fields will be unable to make sense of. What is needed is a description of these computational processes that is not tied to the computational language of some developer's choice. More clarity by way of *idealized* functions in the precise language of mathematics would alleviate this problem.

The rest of this paper is structured as follows. In the next section, we provide a short primer on biology, and subsequently, we examine the structure and contents of the SAM format. The following two sections investigate an appropriate formalization in the context of the questions of extensibility and reproducibility. We then offer concluding remarks.

II. The Underlying Biology

Since we are using bioinformatics data, let us present a few related concepts from Biology. To motivate the reader, let us propose an experiment: we need to explore the differences between the DNA molecules obtained from two individuals of the same species: an 'experimental' individual affected by a disease versus a 'reference' individual without such a disease.

The DNA molecule is essentially a long sequence of nucleotide bases: adenine, guanine, cytosine, and thymine, represented by one of the letters *A, G, C*, and *T* respectively. For many organisms, a reference typically representing the entirety of its genetic material is already published on the web and is commonly referred to as a *genome*.

In the laboratory, there are methods of *DNA sequencing*, i.e., obtaining the exact order of those bases within a DNA molecule extracted from the 'experimental' individual. We will refer to such methods as *wet lab processes*. For organisms with short sequences, i.e., with length in the hundreds, it is a relatively straightforward process. Interesting organisms, however, have very long sequences (the human genome has over 3×10^9 bases). For those, the approach is to fragment the DNA randomly, replicate them, and determine the sequences for the short fragments (for which task there exist machines). The sequences obtained from the wet lab process are referred to as *reads*. Unfortunately, the wet lab methods are error-prone. Hence, the machines emit a probability of correctness (or quality) with each of the bases in the short sequence.

The remaining task is then to attempt to piece them together to infer the original, long sequence. This task is called *alignment*; it is a *dry lab process* in the sense that it is performed using computational methods.

Alignment (or *sequence alignment*, or *sequence mapping*) consists of *matching* each *read* from the experimental DNA against the pre-existing reference sequence. The task is challenging, since each obtained *read* could fit in many locations of the (huge) reference genome. The goal is to find the best match out of all possible matches.

As an analogy, one can think of sequencing as reading a book, attempting to commit its contents to memory, and then matching the memory against the actual contents of the book (reference). As one tries to recall from memory and match against the book, one will most likely realize that one lacks some words, mixes up the order of others, and so on; the result would be a series of 'close' matches, each with a notion of where it might properly fit in the actual book.

For example, suppose one wanted to align the sequence "TAAGCT" with the reference "TACGGT." There is more than one way they might align; two of them (as per the Needleman-Wunsch algorithm [9]) are shown in Tables I and II.

TABLE I. Sequence Alignment

Position	1	2	3	4	5	6	7	8
TACGGT	T	A	_	C	G	G	T	_
TAAGCT	T	A	A	G	_	_	C	T

One possible match between "TACGGT" and "TAAGCT" using Needleman-Wunsch for global alignment. Insertions and deletions are denoted by underscores.

TABLE II. Alternative Sequence Alignment

Position	1	2	3	4	5	6	7	8
TACGGT	T	A	_	C	G	G	_	T
TAAGCT	T	A	A	_	G	_	C	T

Another possible match between "TACGGT" and "TAAGCT" using Needleman-Wunsch for global alignment. Insertions and deletions are denoted by underscores.

In I, the character bases TA match exactly; A is an insert-error (requires it to be inserted into the reference string), G is a mismatch error, then there are two delete errors (the next two characters of the reference GG must be deleted), C is a mismatch error, and finally T is an insert-error.

Part of the task is to decide the best match. One way is to assign scores to each match/error; e.g., each insertion/deletion (*indel*) and mismatch error can be given negative values and matches positive values. Then the sum of the scores for each base in the experimental sequence can reflect the goodness of this match. The *best* match would be the one with the highest overall score.

It is possible to have a more complex scoring matrix in which each type of base match/mismatch/indel has a different value (e.g., A–G mappings might be given a higher value than A–C).

Another possible approach is to take into account the quality values of each base in the sequence obtained from the wet lab experiment. For example, a mismatch error may be forgiven if the base in question was already flagged with a poor quality score [10].

Yet another approach may penalize each *beginning* of a series of gaps (inserts or deletes) since larger runs of gaps are usually more biologically plausible (as opposed to many small insertions/deletions peppered throughout a sequence). Using this approach, the alignment in Table II would yield a higher score than that in Table I, since the former has two runs of two contiguous gaps (at positions 3–4 and 6–7), while the latter has only one contiguous gap (at 5–6), and two individual gaps (at 3 and 8).

The algorithms typically involve some type of dynamic programming [11]. Since this is time consuming for large sequences, modern approaches use heuristics to reduce the time necessary. Two well-known examples are FASTA [12] and BLAST [13].

 ISBN: 978-1-61208-415-2

```
gi|110640213|ref|NC_008253.1|... 16
gi|110640213|ref|NC_008253.1| 611 42 70M *
0 0 TTTCGTCGACCAGGAATTTGCCCAAATAAAACATGT...
22222222222222222222222222222222222... AS:i:0
XN:i:0 XM:i:0 XO:i:0 XG:i:0 NM:i:0 MD:Z:70 YT:Z:UU

gi|110640213|ref|NC_008253.1|... 0
gi|110640213|ref|NC_008253.1| 215 42 70M *
0 0 CCACCCCCATCAGCATTACCACAGGTAACGGTGCGG...
22222222222222222222222222222222222... AS:i:-6
XN:i:0 XM:i:2 XO:i:0 XG:i:0 NM:i:2 MD:Z:5A6C57
YT:Z:UU

gi|110640213|ref|NC_008253.1|... 16
gi|110640213|ref|NC_008253.1| 706 40 70M *
0 0 CCCGTGGCGAGAAAAGGTCGATAGCCATTAGGGCCG...
22222222222222222222222222222222222... AS:i:-12
XN:i:0 XM:i:4 XO:i:0 XG:i:0 NM:i:4 MD:Z:0G14T6C7T39
YT:Z:UU
```

Figure 1. Three alignment lines from a SAM file. Fields are delimited by whitespace. These lines are copied from a larger dataset [16].

- **QNAME:** *gi|110640213|ref|NC_008253.1|...*
- **FLAG:** *16*
- **RNAME:** *gi|110640213|ref|NC_008253.1|*
- **POS:** *706*
- **MAPQ:** *40*
- **CIGAR:** *70M*
- **RNEXT:** ***
- **PNEXT:** *0*
- **TLEN** *0*
- **SEQ:***CCCGTGGCGAGAAAAGGTCGATAGCCAT...*
- **QUAL:** *22222222222222222222222222...*
- **TAG:** *AS:i:-12 XN:i:0 XM:i:4 XO:i:0 XG:i:0 NM:i:4 MD:Z:0G14T6C7T39 YT:Z:UU*

Figure 2. The third alignment line from Figure 1 decomposed into component fields. The ellipsis are used here to denote that field's value continues on. The TAG field itself has several subfields of tags.

A. Interpreting the Alignment

When the locations and lengths of the reads are overlaid, one on top of another at their aligned locations on the reference, the ones with a great deal of overlap (called *pileup*) lead to stronger confidence in the fragment of the genome they cover.

Interestingly, this can lead one to conclude that while most of the genome matches, a part of it definitely has been altered in the experimental sequence. Suppose that one finds that no matter how a specific *read* was aligned, a handful of particular mismatches are manifested consistently. To return to the example we started with in this section, one might conclude that those consistent mismatches exhibit a mutation that contributed to the experimental individual's risk for the disease in question.

III. The SAM Format

The Sequence Alignment Format (SAM) [14] captures the result of both the dry and wet lab processes described in Section II. It is designed to be used with a specific software package, called SAMTools [15]. It uses the alignment information contained in SAM files to perform various taks such as visualizing overlapping reads against the reference and examining pileups in a binned fashion (e.g., how many reads start at a given position of the reference).

The SAM format is similar to that of a Comma-Separated Value (CSV) file with implicit component names (an Extensible Markup Language (XML) equivalent with named fields (components) can easily be conceived).

An optional header section contains general information relevant to the entirety of reads in the file, e.g., the SAM format version number, the specifics of the alignment lines at various locations, and the order in which data will appear.

A. The Alignment Section and Lines

Immediately following the header section (should it exist at all) are the rows of alignment lines corresponding to each *read*. (It should be noted that sometimes the reported sequences correspond to smaller pieces of reads called *read segments*). Predictably, the bulk of the SAM file is taken up by the alignment section, filled with alignment lines.

A sample of three alignment lines from an example SAM file is given in Figure 1. The third alignment alignment line is decomposed into its component fields in Figure 2. There are at least eleven distinct fields, delimited by whitespace, in each alignment line. The twelfth field, TAG, is optional and slightly different, since there can be multiple tags within that field or even none at all (multiple tags are also delimited by whitespace). These are numbered 1 through 12; here and always in discussing the SAM file format, enumeration starts at one (i.e., all sequences are *1-indexed*).

The **SEQ** field contains the sequence corresponding to the read as obtained by the wet lab process.

The **QUAL** field (holding a quality string) captures the probability of incorrectness of each base in that sequence reflecting the error-prone nature of the wet lab processes. Each base is associated with an ASCII character, which can be transformed into a probability of correctness by different functions depending on the scoring method originally used to encode it. One equation to interpret characters of the QUAL field is given in (1).

$$P = 10^{\frac{QUAL-33}{-10}} \tag{1}$$

where QUAL is the decimal value of the ASCII character and P is the probability of the associated base being incorrect. As an example, the ASCII value of "2" is 50. When used with (1), the resulting probability P for its associated base being wrong is 0.0199 (or about 1-in-50).

The **RNAME** field gives the reference. It is typically the value of (or contains the value of) some genetic identifier (e.g., an NCBI genetic code [17]) with a code representing which organism the reference comes from. For instance, in the example SAM line in Figure 2, RNAME is an NCBI code with value *gi|110640213|ref|NC_008253.1|*. The portion "NC_008253.1" is an accession number, giving a value for a particular sequence record. The number "1" after the period says that this is the first version of the sequence, which happens to be the entire genome of *Escherichia coli 536*.) The "gi" code given in "gi|110640213" is another version number [18].

The **POS** field provides the offset (1-indexed) into the reference sequence where the experimental sequence was best matched. In Figure 2, POS has the value *706*, meaning that

 ISBN: 978-1-61208-415-2

the read segment on this alignment line starts at offset 706 of the *Escherichia coli 536* genome according to the best match.

The **CIGAR** field contains a string of the same name, CIGAR (Compact Idiosyncratic Gapped Alignment Report), which SAMTools uses as a coded abbreviation for how one sequence matches to a reference [14]. It contains an ordered series of numbers and letters, which tells one precisely how the sequence maps. Numbers correspond to the length of a subsequence while the letters imply an edit. An example of such mapping is given in Table III.

Referring back to Table I, its CIGAR string would read: $2M1I1M2D1M1I$. It would mean a match obtained in the following way:

> *The first two characters match or mismatch, the next character was missing in the reference and had to be inserted; the next character is a match or mismatch; the next two characters had to be deleted from the reference; this is followed by a match or mismatch; and the last character is an insertion into the reference.*

In Figure 2, the CIGAR field reads "70M", meaning the read aligned to the reference sequence with some combination of 70 matches and mismatches. Some of the extended CIGAR codes and their meanings are given in Table IV.

TABLE III. SEQUENCE ALIGNMENT WITH CIGAR

Position	1	2	3	4	5	6
AACTG	A	A	C	T	G	_
ACTGG	A	_	C	T	G	G

Best match between the two strings "AACTG" and "ACTGG" using Needleman-Wunsch for global alignment. Insertions and deletions are denoted by an underscore. The CIGAR string for "ACTGG" would read 1M1D3M1I.

TABLE IV. CIGAR STRING CODES

Op. Letter Code	**Meaning**
M	Alignment Match (sequence match or mismatch)
I	Insertion to the reference
D	Deletion from the reference
N	Skipped region from reference

Some (but not all) lettered operation codes and their meanings present in the extended CIGAR format.

The **TAG** field can contain many tags (minimum zero), even user-specified ones. All tags have the format of NAME:TYPE:VALUE and each can only appear once in any given alignment line [14]. 44 unique tags are established in the current SAM format, and each provides additional information from textual comments to the original alignment score generated by some aligner. (These tags are essentially additional pieces of metadata.)

The **MAPQ** field gives a mapping quality (higher number implies higher quality) for the alignment which, as we have stated earlier, is a difficult task. The MAPQ value of the SAM line given in Figure 2 is 40, denoting a high-quality match.

IV. FORMALIZING SAM-FORMATTED DATA

SAM data is very useful: it says a lot about what occurred in sequence alignment and also something about sequencing. It is possible, however, that the software used to generate the data could change with time. For example, the change could be in an underlying alignment algorithm or the interpretation of read quality values. It would be beneficial if one could take such improved data and compare it with older SAM data. We argue that a formalization that shows *how* to represent and compute information would address this problem.

Such a formalization should have three parts: an ontology and two kinds of functions; we describe them below.

Such a formalization should live alongside the current data as supplementary metadata. In this way, all data can be understood more readily by domain experts while also revealing the specific functionality of that data (e.g., how to interpret an alignment sequence's structure using the CIGAR field).

A. Ontology

We need an ontology that codifies concepts of the knowledge domain appropriate to the data. The ontology needs to contain the domain concepts arranged (possibly) in a generalization/specialization hierarchy; plus mappings among them; The full ontology necessary to capture all the semantic concepts of a SAM file would be far too large to present here. Instead, we furnish the reader with those essential constructions that clarifies the point of formalization: some of those entities and relationships that are components of the *sequence alignment* concept. This subset of semantic concepts are given in Figure 3.

The ontology in Figure 3 is the minimal amount needed to describe the concepts involved in sequence alignment. A parent class for many of the concepts here is *Sequence*, which is why many other concepts, such as experimental and reference sequences are its child classes. Any concept that has *Sequence* as a parent in the hierarchy will inherit from its definitions. For example, anything that is a *Sequence* will have a "character_seq_string" (the actual string representation of characters in the sequence).

ExperimentalSequence is distinct from *ReferenceSequence* (though they are both child classes of Sequence). *ExperimentalSequence*s have both quality scores (denoted by the "quality_scores_string" attribute) and is composed of at least 1 *Read* (with the *minimum 1* cardinality restriction). *ReferenceSequences*, instead, have an *OrganismName* attribute.

Reads, as we have said, can be split up into *ReadSegment*s, and these *ReadSegment*s form a chain from one *ReadSegment* to the next (the *hasNextReadSegment maximum 1* cardinality restriction reflects this). A *Read*, then, is composed of at least one *ReadSegment*, and *ReadSegment*s are components of *Read*s.

SequenceAlignment is the concept related to actually doing something with these various sequences. It acts as a function signature for performing sequence alignment in general. It has two inputs listed, which are an *ExperimentalSequence* and *ReferenceSequence*. The output is *minimum 0 AlignedSequence*s, because it is possible no alignment can be found. This *minimum 0* also means that many *AlignedSequence*s can exist for some SequenceAlignment. The "FunctionInternal" attribute is explained in Subsection IV-B, and describes "how" this *SequenceAlignment* concept computes its output.

The *AlignedSequence* is the output of some *SequenceAlignment* (as clear in the *isOutputOf* relation). This concept has an

Sequence (1)

Attribute character_seq_string (String)
Attribute length (Integer)

ExperimentalSequence (2)

Attribute quality_scores_string (String)
isA Sequence
isComposedOf *minimum 1* Read

ReferenceSequence (3)

Attribute OrganismName (String)
isA Sequence

Read (4)

isA Sequence
isComposedOf *minimum 1* ReadSegment
isComponentOf *minimum 1* ExperimentalSequence

ReadSegment (5)

isA Sequence
isComponentOf Read
hasNextReadSegment *maximum 1* ReadSegment

Sequence Alignment (6)

Attribute FunctionInternal (XML Literal)
hasOutput *minimum 0* AlignedSequence
hasInput *some* ExperimentalSequence
hasInput *some* ReferenceSequence

AlignedSequence (7)

Attribute ReferenceOffset (Integer)
isA Sequence
mapsTo *exactly 1* ReferenceSequence
isOutputOf *some* SequenceAlignment

Figure 3. A subset of the full formalization of the SAM format to which actual data will be mapped either directly or through complex transformation.

attribute for stating the offset in which the alignment occurs against *exactly 1 ReferenceSequence* (the reference used in sequence alignment).

B. Choice of Ontology

The snippet of the ontology in Figure 3 is based on the "EMBRACE Data And Methods" (EDAM) ontology [19]. EDAM was based on several different resources, including myGRID [20], and follows many principles of the Open Biological and Biomedicial Ontologies. EDAM attempts to represent formats used in bioinformatics, their data, the operations those data might be involved in, and associated topics. The EDAM ontology is impressive in its coverage and design. There is, to our knowledge, no more comprehensive ontology than EDAM that exists with concepts tying both bioinformatics formats to their data and use in associated operations.

We, however, do not find EDAM capable of fulfilling all the needs of our goal here, since we require more complete coverage of individual formats with more format-specific constraints. When trying to fit more specifics of SAM data with what is presented in EDAM, we found some concepts to be missing. This is most likely because EDAM was designed to represent workflows, which is at a higher level of granularity than the specifics of a data format's contents. These missing concepts include a quality sequence for SAM's QUAL field and a concept appropriate for mismatching character sequences such as CIGAR strings. Further, we wanted to be able to pair individual portions of an alignment line with one another, which required more specifics relating to the SAM file.

To be clear, in some ontologies, such functions may be represented as a concept, but it is in name and relationships only; they are function signatures, but do not contain function internals. The myGrid ontology, for example, contains the concept of the Needleman-Wunsch sequence alignment algorithm. The EDAMS ontology has the more abstract *SequenceAlignment* concept. While these concepts may contain relationships such as output, input, etc., they lack a conceptualized view of the algorithm — or process — they are meant to define. In other words, there is no implementation-agnostic function internals to pair with real-world instances.

In Figure 3, we show our plans for adding such functional internals to ontological definitions. This "FunctionInternal" attribute (found in the *SequenceAlignment* concept) of type XML literal can be represented with MathML [21]. It is with this language that the implementation-agnostic function internals can be described.

C. Functions and Extensibility

Consider Figure 4, a commutative diagram [22] connecting the worlds of data and ontology via functions. Functions of the first kind are like the dashed arrows in that figure. They take one or more data elements and bridge them to a concept in the ontology [23]. Our second kind of functions are like $F(x)$ and $F'(x)$. They represent some idealized computation of some domain concepts from others, reflecting data elements that are computed from other data elements in an equivalent manner.

For example, let *Data* map to a pair of read and reference, *Ontology Term 1* to a pair of sequences, *Results* to an alignment with an offset, and *Ontology Term 2* to an aligned sequence. Let $F(x)$ be the Needlemann-Wunsch algorithm and $F'(x)$ be a particular implementation of that algorithm. It is easy to see the power of this framework; for example, it would provide the ability to use more complex alignment algorithms that take into account the probabilistic nature of the sequences under consideration.

One synergistic consequence is extensibility. One can take existing data and compare against the results of newer alignment algorithms (e.g., BLAST and FASTA, or algorithms being researched).

V. Investigation of Reproducibility of Results

We asked the question: can the final result of the dry lab process, the MAPQ value, be reproduced by a consumer of a SAM data file?

 ISBN: 978-1-61208-415-2

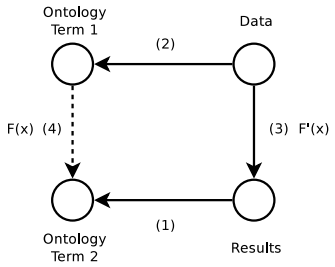


Figure 4. Commutative diagram for data and ontology

A. Computing Alignment Score

As mentioned earlier, the SAM file includes a MAPQ field which contains a number indicating what confidence one may have in the alignment inferred by the dry lab process. The value is given by (2), where *P* is the probability of error of the stated alignment of read z at offset u in genome x:

$$MAPQ = -10\log_{10}(P) \tag{2}$$

P is given by

$$P = 1 - \frac{p(z|x,u)}{\sum_v p(z|x,v)} \tag{3}$$

where $p(z|x,u)$ is the probability that read z maps to reference x at offset u [24]; in the denominator, the summation is over all such offsets.

The *best match* gives the offset u that results in the smallest P as per (3). This is the location offset with respect to the reference where the best alignment match can be found. Equation (3) implies an exhaustive computation (albeit there are positions that can be skipped over).

However, there are more modern, less time-consuming alignment methods based on heuristics. For example, the following approximation is faster [24].

$$\begin{aligned} MAPQ = min[q_2 - q_1 - 4.343\ log(n_2), \\ 4 + (3 - k')(\bar{q} - 14) - 4.343\ log(p_1)\ (3 - k', 28)] \end{aligned} \tag{4}$$

where q_2 is the sum of the quality scores corresponding to the bases that have mismatches for the best alignment score and q_1 the corresponding sum for the second-best alignment. k' is the minimum number of mismatches on some 28 base-pair *seed* (which are locations to index against the reference with the read; larger seeds result in lower sensitivity [25]) and $\bar{q}$ is the average of base qualities in that seed [24].

Clearly then, the MAPQ computed depends on the equation used; more generally, if context-sensitive cases are considered, on the algorithm used. This means that different programs used in alignment of the same experimental sequence against the same reference may result in different values of MAPQ. Without knowledge of the equation or algorithm used, one cannot reproduce the MAPQ element presented in the SAM file. Worse, comparing MAPQ values of multiple SAM files may well be invalid. (Adding a pointer to a program used for alignment is possible, but such additions would be in optional fields with no guarantee of inclusion in all published data files.)

For reproducibility, we suggest that the function (or algorithm) used in the dry lab process for MAPQ be included in the SAM data file.

This will allow users not only to verify the correctness of the MAPQ reported in the file but also use their own alignment algorithm to compare and publish the results obtained. This could lead to publications providing insight into various aspects of experiments such as innovative approximations and the effect of errors in the underlying wet lab process.

VI. Conclusion and Future Work

Examining a popular file format in bioinformatics, we asked two questions relating reproducibility and extensibility of data. What we found was that neither reproducibility nor alternative conclusion testing was possible. We identified the cause as the absence of functions for arriving at these conclusions.

Such functions are essential extensions to whatever current metadata may be included with the data, since these functions expose the details of how the data was created in the first place. When such functions exist, one can start to meaningfully compare different data sets that did not employ the same function and also to compare the computed results (e.g., MAPQ) in a data set with the corresponding results that one would obtain using an alternative function or algorithm.

We have also explored the choice of what metadata to add and how to add it. We have shown that this choice need not be guided by a particular pre-defined purpose, but from the contents of the data itself. When data is formalized, ambiguity and confusion about how different parts of data relate to one another are decreased. Necessary metadata becomes apparent as the portions of data are formalized and their component parts are established.

This has culminated in the idea of the formalization of data as a marriage between an ontology and a set of functions that describe both data and its underlying processes. Such a formalization creates an accessible point from which data can be understood, reproduced, and have its functionality extended.

While we have explored an example from bioinformatics and presented a subset of the constructions required for a formalization, we have tried to show that the generality of the approach goes beyond bioinformatics and embraces any data file containing computed elements. Such data files are pervasive; consider for example, a medical report such as a blood test that contains diagnostic elements that are based on complex analysis.

Of course, our work is part of a larger solution. For example, the decision of how best to represent such additional metadata is left as further research.

Acknowledgment

We would like to thank anonymous reviewers whose comments have helped improve this paper. This work is supported by a grant from The United States Department of Homeland Security 2011-ST-062-000051. We gratefully acknowledge travel support from the Institute for Complex Additive Systems Analysis (ICASA) at New Mexico Tech enabling the presentation of this paper.

 ISBN: 978-1-61208-415-2

REFERENCES

[1] V. Stodden, "Reproducible research: Addressing the need for data and code sharing in computational science," Computing in Science & Engineering, vol. 12, no. 5, 2010, pp. 8–12.

[2] R. LeVeque, I. Mitchell, and V. Stodden, "Reproducible research for scientific computing: Tools and strategies for changing the culture," Computing in Science and Engineering, vol. 14, no. 4, 2012, p. 13.

[3] R. Stevens, A. Robinson, and C. Goble, "MyGrid: personalised bioinformatics on the information grid," Bioinformatics, vol. 19, no. suppl 1, 2003, pp. i302–i304.

[4] C. Pancerella et al., "Metadata in the collaboratory for multi-scale chemical science," in International Conference on Dublin Core and Metadata Applications, 2003, pp. pp–121.

[5] T. Oinn et al., "Taverna: a tool for the composition and enactment of bioinformatics workflows," Bioinformatics, vol. 20, no. 17, 2004, pp. 3045–3054.

[6] J. Frew and R. Bose, "Earth System Science Workbench: A data management infrastructure for earth science products," in Scientific and Statistical Database Management, 2001. SSDBM 2001. Proceedings. Thirteenth International Conference on. IEEE, 2001, pp. 180–189.

[7] D. Wood, M. Lanthaler, and R. Cyganiak, "RDF 1.1 concepts and abstract syntax," W3C, W3C Recommendation, February 2014, [Retrieved May, 2015]. [Online]. Available: http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/

[8] P. Patel-Schneider and B. Motik, "OWL 2 Web Ontology Language mapping to RDF graphs (second edition)," W3C, W3C Recommendation, December 2012, [Retrieved May, 2015]. [Online]. Available: http://www.w3.org/TR/2012/REC-owl2-mapping-to-rdf-20121211/

[9] S. Needleman and C. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," Journal of Molecular Biology, vol. 48, no. 3, 1970, pp. 443–453.

[10] X. Yu et al., "How do alignment programs perform on sequencing data with varying qualities and from repetitive regions?" BioData Mining, vol. 5, no. 1, 2012, p. 6.

[11] O. Gotoh, "Multiple sequence alignment: algorithms and applications," Advances in Biophysics, vol. 36, 1999, pp. 159–206.

[12] W. Pearson and D. Lipman, "Improved tools for biological sequence comparison," Proceedings of the National Academy of Sciences, vol. 85, no. 8, 1988, pp. 2444–2448.

[13] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman, "Basic local alignment search tool," Journal of Molecular Biology, vol. 215, no. 3, 1990, pp. 403–410.

[14] "Sequence alignment map format specification," May 2013, [Retrieved May, 2015]. [Online]. Available: http://samtools.sourceforge.net/SAMv1.pdf

[15] H. Li et al., "The Sequence Alignment Map format and SAMtools," Bioinformatics, vol. 25, no. 16, 2009, pp. 2078–2079.

[16] E. Cerami, "SAMtools: Primer," [Retrieved May, 2015]. [Online]. Available: http://biobits.org/samtools_primer.html

[17] A. Coghlan, "Sequence Databases," [Retrieved May, 2015]. [Online]. Available: http://a-little-book-of-r-for-bioinformatics.readthedocs.org/en/latest/src/chapter3.html

[18] B. Hochhut et al., "Escherichia coli 536, complete genome," [Retrieved May 2015]. [Online]. Available: http://www.ncbi.nlm.nih.gov/nuccore/NC_008253

[19] J. Ison et al., "EDAM: an ontology of bioinformatics operations, types of data and identifiers, topics and formats," Bioinformatics, vol. 29, no. 10, 2013, pp. 1325–1332.

[20] J. Zhao et al., "Using semantic web technologies for representing e-science provenance," in The Semantic Web–ISWC 2004. Springer, 2004, pp. 92–106.

[21] R. Miner, "The importance of MathML to mathematics communication," Notices of the AMS, vol. 52, no. 5, 2005, pp. 532–538.

[22] J. Adamék, H. Herrlich, and G. Strecker, Abstract and Concrete Categories. John Wiley, 1990.

[23] J. McCloud and S. Mazumdar, "Translation of Various Bioinformatics Source Formats for High-Level Querying," in 2013 Linked Data in Practice Workshop (LDPW2013), 2014, pp. 39–53.

[24] H. Li, J. Ruan, and R. Durbin, "Mapping short DNA sequencing reads and calling variants using mapping quality scores," Genome Research, vol. 18, no. 11, 2008, pp. 1851–1858.

[25] J. Buhler, "Efficient large-scale sequence comparison by locality-sensitive hashing," Bioinformatics, vol. 17, no. 5, 2001, pp. 419–428.

 ISBN: 978-1-61208-415-2

Automatically Triggering Activity and Product Predictions in Mobile Phone Based on Individual's Activity

Kalpana Algotar
Retail Solution Division
Intel Corporation
Phoenix, USA
e-mail: Kalpana.a.algotar@intel.com

Sanjay Addicam
Retail Solution Division
Intel Corporation
Phoenix, USA
e-mail: Sanjay.v.addicam@intel.com

***Abstract*—The technological advances in mobile phone and their widespread use has resulted in the big volume and varied types of mobile data we have today. Researchers have begun to mine mobile data in order to predict a variety of social, economic, personal, location and health related events. Mobile data directly reflects individual's life without disclosing personal information, and therefore it is an important source to analyze and understand the underlying dynamics of human behaviors or activities. In this paper, we describe an innovative and challenging process to predict user's activity using mobile based data. We propose a graph-based framework that uses the user's activities, social network, and product-keywords in order to provide recommendations which are also delivered through mobile phones. This paper summarizes the different types of prediction logic algorithms by constructing graphs from different data sources. Our graph-based approach is highly scalable and can be used to predict individual's next activity, as well as prediction towards products purchase. The mobile recommendation engine incorporates three types of data to generate the graph and to predict activity and product. First, we collect product-keywords using text-rank algorithm. Second, we collect individual mobile's past data, such as accelerometer, call log, battery status, app usage, browsing history, Facebook data, and Twitter data. Third, we collect user's mobile phone activity 8 times during the day. By using multimap, we get fast prediction in real-time mobile.**

Keywords-Text-Rank Algorithm; Multimap; Adjacency List; Internal Prediction; External Prediction.

I. INTRODUCTION

The knowledge of user activities and habits is a crucial factor for the development of highly personalized applications that can be beneficial in many areas of daily life [3]. The mobile users' behaviors (e.g., SMS, call history, location, app usage, battery status, accelerometer Facebook, Twitter, etc.) are all related to real-world behaviors. This provides an unprecedented opportunity for us to understand the underlying dynamics of users' behaviors in the mobile data [2]. In this work, we aim to answer an interesting question, i.e., whether we can predict a user's next activities based on his/her historic behavior log/activities, such as call log, browser history, app usage and mobile social network information.

In this paper, we explore and develop novel methods for recognition of user's next activities. Mobile devices present an ideal platform for this task; they usually possess considerable computational resources, a rich set of wireless communication and multimedia features [3]. Nowadays, people tend to always carry their phones along. We employ a mobile phone as the main sensory and processing unit for learning and predicting user's behavior [3].

Recently, considerable related works have been conducted, e.g., activity recognition [4]-[9], dynamic emotion analysis [10]-[14], dynamic social network analysis [15]-[19], and social influence analysis [20]-[24]. Emotion analysis is to study how an individual's emotional state (e.g., happiness and loneliness) propagates through social relationships [10]-[12]. Dynamic social network analysis is to model how friendships drift over time using a dynamic model [18] or to investigate how different pre-processing decisions and different network forces such as selection and influence affect the modeling of dynamic networks [19].

This paper proposes a Mobile Recommendation Engine/Framework (MRF), which builds graph using different data sources, and applies different types of prediction logic using multimap structure which enables to get user's next activity e.g., walking, calling, eating etc., as well as product prediction that user is interested to see or purchase. Multimap is a generalization of a map or associative array abstract data type [25]. The reason behind using multimap is that it allows storing multiple values for every key, as well as, it allows storing duplicate keys. It has useful methods, i.e., invertFrom, which copies each key-value mapping in source into destination, with its key and value reversed without using loop. For backtracking logic, we traverse the path in the reverse direction until we hit the root node that is not have any parent node. In this case, consider every node in the reverse path as a value and find the attached keys for each corresponding values using invertForm method of multimap. Here, we present a different method using user's context (or activity) in mobile to predict user's next action.

The other sections in this paper are organized as follows. Section 2 illustrates the source of data. Section 3 describes our approach. Section 4 presents the process for deploying the experiment and the results of the process. Section 5 concludes the paper.

 ISBN: 978-1-61208-415-2

II. SOURCE OF DATA

In our experiment, we used the following different data sources.

A. Historical Data

We have (i) historical data, such as call log, battery status, app usage, browsing history, accelerometer data, and (ii) social network data, such as Twitter, Facebook data based on individual's past activities on mobile, etc. We have used this information without revealing individual's personal information.

B. Product Data

We crawled the Amazon site and collected around 750 products reviews using an HTML-embedded scripting language (PHP) script. We generated top keywords from those reviews using Text-Rank algorithm [27]. This algorithm provides unique keywords along with weight for each product. The Text-Rank algorithm helps to distinguish each product based on unique keywords.

C. User's Behaviour Data

Every day, we collect user's activities on mobile, such as call log, app usage, browser history, accelerometer data, social network data, such as Facebook, and Twitter at three hours intervals and build subgraph. After experimenting on various hours interval, we found that three hours interval is sufficient to collect the required amount of activities to build subgraphs.

III. OUR APPROACH

To build a mobile framework, we collect different types of data, such as historical data, product-keyword data, individual's activity on mobile, and build 200,000 nodes on a graph. We stored this graph in cloud because the size of the graph is big and one can send graph to individual's mobile phone based on his/her matching profile. We used graph cut algorithm on main graph that generates many subgraphs. Then, we used Approximate Subgraph Matching (ASM) [28] algorithm to pick the subgraph that is more relevant to individual's activity. This relevant subgraph is loaded on individuals' mobiles every 24 hours. We build backtrack and product prediction logic on subgraph that is loaded in mobile and based on last 3 hours user activities used to give future prediction.

IV. EXPERIMENTS

Graph technology is the process of analyzing large volume of data from different perspectives and summarizing it into useful information – information that can be used for prediction. Here, we developed three types of prediction logic to get three types of predictions. We implemented different graph algorithms, to build, merge, cut, and compare the graph, and to get prediction for individual's activity and product preference. First, we collect the data from the different sources. Second, we defined relationships between different data sources and put those data as nodes and edges on graph, as presented in Figure 1. For example, user node is connected with different activity nodes, such as call log, app usage, accelerometer, and browsing history. We extracted keywords from browsing history and that node is connected with extracted keywords and keyword nodes are connected with product nodes. We placed this big constructed graph on Content Management Server (CMS). Third, we collect user's activity every 3 hours and represent those activities as a subgraph. At the end of the day, we have 8 subgraphs. Fourth, using graph merge algorithm, we merge 8 subgraphs into one subgraph. Fifth, we compare this merged graph with main graph and put nodes on main graph which are not exist in main graph. In this stage, we update the main graph on CMS using merge graph. Sixth, using graph-cut algorithm, we cut the main graph into no. of subgraph. Lastly, using ASM algorithm, we compare the merge graph from step-4 with subgraph from step-6 based on node, edge weight, edge, direction and most likely matched subgraph send to that user's mobile from CMS every 24 hours. We are calculating edge weight in two ways. One is occurrence and the other is time difference. Occurrence is calculated based on repetition. For example, after outgoing_3, walking is done. This action is repeated 6 times, then the occurrence edge weight is 6. If the time difference between the call and walk is 45 minutes, then the time difference edge weight is 45 minutes. Next time, if the time difference is 30 minutes, the edge weight is updated as the average of 30 and 45 minutes. The 3rd time, if the time difference is 20 minutes then the edge weight is calculated as 20+30+45/3. For predicting user activity and product preference, we utilize user activities for the last 3 hours as an input.

A. Internal Prediction

We collect the last 3 hours of the user's activities from individual's mobile and use them as an input on subgraph, as shown in Figure 1. We search each activity as value on subgraph. If value exists on subgraph, then, we find the key for value recursively until we hit the orphan node that is not having any parent node. For example, we have two user actions 5 and 6 in the last 3 hours. First, take user action 5 as an input value and find the attached keys as 4 on subgraph and store key 4 as a key node into collection hashmap-A. Hashmap is a data structure used to store object and retrieve it in constant time O(1). It works on hashing principle in Java. Now, take one-by-one node as a value from hashmap-A and find the attached key as 3 for value 4 and append the key node into hashmap-A and we will have keys {4, 3} and marked node 4 as processed. We repeat the same process for unprocessed nodes of hashmap-A and append the result to hashmap-A. At the end, the result for user activity 5 will be {4, 3, 2, 1}. In this case we stop at node 1, because no edge is coming into it. Likewise, take the second user activity, say 6, and follow the same process as mentioned for node 5 and store the key of each recursive operation into new collection hashmap-B. The result for user activity 6 will be {2, 1}. At the end, we intersect two collections hashmap-A {4, 3, 2, 1} and hashmap-B {2, 1} and store the intersection result {2, 1} into the collection hashmap-C.

If more than two user actions exist, then, we follow intersection between first and second, second and third, third and fourth, and so on, and append each intersection result into

 ISBN: 978-1-61208-415-2

collection hashmap-C. Then, we apply Depth First Search algorithm [29] to find the path between source and destination. We take the first node, say 2, as a source from hashmap-C, and the first user activity from last 3 hours, say 5, as a destination and pass these data/nodes as parameters to Depth First Search algorithm. If a path exists between these two nodes, then, we give source node 2 as an internal prediction. If a path does not exist between source and destination, then, we choose second node 1 as source from hashmap-C and user activity 5 as destination and pass these values to depth-first search algorithm. We follow the above process recursively, until we find a path.

B. External Prediction

In this stage, we take one by one user's activities from last 3 hours storage and find on subgraph that is loaded in mobile every 24 hours. Here, we consider user action 5 and 7 as keys and find the corresponding attached values, say {8, 9, 10} and {11, 12}. Then after, we move in the forward direction with one degree depth. If a user activity exists in the subgraph Figure 1, and has more than one outgoing edge, then, choose the highest edge weight node that is connected with the user activity. We follow the same process for each user activity node, and, at the end, we select the outgoing edge with maximum weight as external prediction among all user activity's highest edge weight node. In this case, we select the node with highest edge weight as an external prediction out of other nodes {8, 9, 10, 11, 12}.

C. Product Prediction

In this stage, we created separate adjacency list for product-keywords in which a keyword acts as a key and products act as a value. We check if the type of user activity is keyword, then, we follow the product prediction logic using product-keywords adjacency list. We can consider that user activities 2 and 4 are keywords and find those nodes as keys on subgraph which is defined in Figure 1, and count how many edges come out from each keyword (user activity node). From Figure 1, we see that two edges are coming out from user activity 2 and three edges are coming out from user activity 4. We follow the same process for all user activity nodes whose type is keyword. Suppose that more than one user activity has type keyword; then, we choose the product as prediction based on the maximum number of keywords (user activity) connected with the same product. In our case, user activities 2 and 4 are both connected with product node 6. It means product node 6 has count 2, while others product for example 5, 7, and 3 has count 1. In this case, we give node 6 as a product prediction, because of a bigger number of user activities with type keyword is connected with product type node 6.

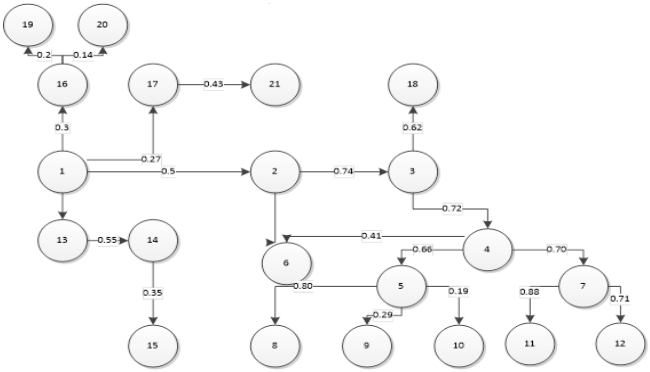


Figure 1. Subgraph in Mobile.

V. CONCLUSION

This paper proposed a novel backtracking approach of recognizing next individual's activities using individual's personal data across a plurality of users' data. This approach is better because multimap has useful utilities like invertForm, which helps to give accurate prediction in a real-time without using a supervised classification algorithm. Multimap, allows multiple values for every key, such as keyword power as a key, can map with multiple products as values, such as mobile, TV, dishwasher, etc. We can expand multimap dynamically as needed. In addition, multimap is lightweight component as it uses less memory.

Briefly described, the individual's interest disclosure pertains to personal data mining. More specifically, data mining technologies can be applied to personal user data provided by users themselves, gathered by others on their behalf and/or generated and maintained by third parties for their benefit or as required [26]. Mining of such data can enable identification of opportunities and/or provisioning of recommendations to increase user productivity and/or improve quality of life [26]. Further yet, such data can be afforded to businesses involved in market analysis, or the like, in a manner that balances privacy issues of users with demand for high quality information from businesses [26].

In accordance with an aspect of this disclosure, personal user data can be received or otherwise acquired from individual's mobile [26]. Graph techniques can be applied to the personal data across a plurality of user, for example, to identify patterns, relations and/or correlations amongst the data. Subsequently or concurrently, mining results and/or useful information based thereon can be provided to a user. We tested the recommendation engine with 25 users for a period of 6 months. We provided 25 users with Motorola G phones with voice plans and data plans and asked them to use these phones as their primary phone. We provide them recommendation every 3 hours and asked them feedback every week on the relevance of the feedback and overall experience. We provided them feedback in terms of Products, Actions and apps. There are no results yet since we are analyzing all the data and feedback collected in the last 6 months and will be producing that as another paper shortly.

 ISBN: 978-1-61208-415-2

REFERENCES

[1] Gomes, B., Phua, C. C. and Krishnaswamy, S. (2013), "Where Will You Go? Mobile Data Mining for Next Place Prediction," International Conference on Data Warehousing and Knowledge Discovery - DaWaK 2013, Prague, Czech Republic.

[2] J. Gong, J. Tang "ACTPred: Activity Prediction in Mobile Social Networks," tsinghua science and technology, issn 1007-0214 05/11 pp. 265-274, vol. 19, no. 3, June 2014.

[3] A. Papliatseyeu and O. Mayora, "Mobile Habits: Inferring and Predicting User Activities with a Location-Aware Smartphone," 3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008, pp. 343-352.

[4] S. Abdullah, N. D. Lane, and T. Choudhury, "Towards population scale activity recognition: A framework for handling data diversity," in Proc. AAAI'12, Toronto, Canada, 2012, pp. 851-857.

[5] J. Yin, Q. Yang, D. Shen, and Z. N. Li, Activity recognition via user-trace segmentation, ACM Transactions on Sensor Networks, vol. 4, no. 4, pp. 19-34, 2008.

[6] D. H. Hu and Q. Yang, Cigar: Concurrent and interleaving goal and activity recognition, in Proc. AAAI'08, Chicago, USA, 2008, pp. 1363-1368.

[7] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, Activity recognition using cell phone accelerometers, SIGKDD Explor. Newsl., vol. 12, no. 2, pp. 74-82, Mar. 2011.

[8] E. Kim, S. Helal, and D. J. Cook, Human activity recognition and pattern discovery, IEEE Pervasive Computing, vol. 9, no. 1, pp. 48 53, Jan.- Mar. 2010.

[9] Y. M. Zhang, Y. F. Zhang, E. Swears, N. Larios, Z. H. Wang, and Q. Ji, Modeling temporal interactions with interval temporal bayesian networks for complex activity recognition, IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 10, pp. 2468-2483, Oct2013.

[10] J. H. Fowler and N. A. Christakis, Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the framingham heart study, British Medical Journal, vol. 337, no. a2338, pp. 1-9, Dec. 2008.

[11] J. T. Cacioppo, J. H. Fowler, and N. A. Christakis, Alone in the crowd: The structure and spread of loneliness in a large social network, Journal of Personality and Social Psychology, vol. 97, no. 6, pp. 977-991, Dec. 2009.

[12] J. Tang, et al., Quantitative study of individual emotional states in social networks, IEEE Trans. Affe. Comp., vol. 3, no. 2, pp. 132-144, April-June 2012.

[13] J. Jia, S. Wu, X. H. Wang, P. Y. Hu, L. H. Cai, and J. Tang, Can we understand van Gogh's mood? Learning to infer affects from images in social networks, in Proc. ACM MM'13, Barcelona, Spain, 2013, pp. 857-860.

[14] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins, Information diffusion through blogspace, in Proc. WWW'04, NewYork, USA, 2004, pp. 491-501.

[15] J. Kleinberg, Temporal dynamics of on-line information streams, in Data Stream Managemnt: Processing HighSpeed Data. New York, USA: Springer, 2005.

[16] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins, Microscopic evolution of social networks, in Proc. KDD'08, Las Vegas, USA, 2008, pp. 462-470.

[17] P. Sarkar and A. W. Moore, Dynamic social network analysis using latent space models, ACM SIGKDD Explorations Newsletter, vol. 7, no. 2, pp. 31-40, Dec. 2005.

[18] J. Scripps, P. N. Tan, and A. H. Esfahanian, Measuring the effects of preprocessing decisions and network forces in dynamic network analysis, in Proc. KDD'09, Paris, France, 2009, pp. 747-756.

[19] A. Anagnostopoulos, R. Kumar, and M. Mahdian, Influence and correlation in social networks, in Proc. KDD'08, Las Vegas, USA, 2008, pp. 7-15.

[20] D. Kempe, J. Kleinberg, and E. Tardos, Maximizing the spread of influence through a social network, in Proc. KDD'03, Washington, DC, USA, 2003, pp. 137-146.

[21] J. Tang, J. M. Sun, C. Wang, and Z. Yang, Social influence analysis in large-scale networks, in Proc. KDD'09, Paris, France, 2009, pp. 807-816.

[22] J. Tang, S. Wu, and J. M. Sun, Confluence: Conformity influence in large social networks, in Proc. KDD'13, Chicago, IL, USA, 2013, pp. 347-355.

[23] J. Zhang, J. Tang, H. L. Zhuang, C. W. K. Leung, and J. Z. Li, Role-aware conformity influence modeling and analysis in social networks, in Proc. AAAI'14, Quebec, Canada, 2014, pp. 1-7.

[24] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri, Feedback effects between similarity and social influence in online communities, in Proc. KDD'08, Las Vegas, USA, 2008, pp. 160-168.

[25] http://en.wikipedia.org/wiki/Multimap

[26] H. Johannes, "Personal Data Mining," USPTO - US7657493, Apr 2008.

[27] https://github.com/samxhuan/textrank/tree/master/src/com/sharethis/textrank (Last access on 06/09/2015)

[28] http://sourceforge.net/projects/asmalgorithm/files/asm/ (Last accessed on 06/09/2015)

[29] http://en.wikipedia.org/wiki/Depth-first_search (Last accessed on 06/09/2015)

 ISBN: 978-1-61208-415-2

Recommender Systems for Museums: Evaluation on a Real Dataset

Ivan Keller, Emmanuel Viennet

L2TI Institut Galilée
Université Paris 13, Sorbonne Paris Cité
F-93430, Villetaneuse, France
Emails: (ivan.keller, emmanuel.viennet)@univ-paris13.fr

***Abstract*—This paper discusses the evaluation of several recommendation methods used to suggest relevant contents to museum visitors. We employed traditional recommender systems along with our versatile Social Filtering formalism to test different strategies on a genuine dataset, which was collected during a recent cultural exhibition that received significant interest in Paris, France. The results show the promising potential of recommendation techniques in the not so well explored application domain of museum visit. This work is part of the AMMICO ongoing research project that aims to develop "smart" audio guides for museums.**

Keywords–Recommender Systems; Social Networks; Museum.

I. INTRODUCTION

Museum visitors are often offered a wide selection of artworks. Most frequently, curators design exhibitions with a linear narrative, and wearable audio guides are optionally available to provide information to the visitors. This setting is generally very static with almost no interaction with the user. Gradually, some museums have developed devices offering predefined suggested visit paths with adapted contents according to the type of the audience, e.g., children, families or school groups. The AMMICO project [1] takes one step further: it aims to provide an audio guide prototype with several novel functions exploiting advanced digital information techniques to enhance the visitor's experience.

The most important functionality on this audio guide is the online recommender system, based on the analysis of the visitor behavior: trajectory (measurement of the accurate position of the user, time dedicated to each artwork), interaction with the device ("likes", search for complementary informations) [2]. In the present study, we will focus on the "likes": the visitor explicitly tells the audio guide that he is interested by the current *Point of Interest* (POI) he is viewing. Building such a recommender system faces the well-known challenges: cold start, data sparsity, over-specialization [3]. Recently, we developed a generic formalism that integrates various classic Recommender Systems (RSs) while providing additional novel ways to implement recommendation [4]: the Social Filtering framework (SF). This versatile tool provides an efficient way to test the performances of many different recommendation strategies. We used SF beside other RSs methods in the museum context. This paper analyses the results we obtained on a real dataset collected during a five-month exhibition held by an AMMICO museum partner. Our contribution lies in revealing the promising potential of recommendation techniques in the not so well explored application domain of museum visit.

The paper is organized as follows: Section II summarizes the concepts and notations of SF while Section III briefly explains the operation mode of traditional RSs we also tested; Section IV recalls the evaluation indicators we used to assess the performances of the tested RSs; Section V describes the dataset on which we ran our experiments; the results are displayed and commented in Section VI. Lastly, our conclusion identifies issues and perspectives.

II. SOCIAL FILTERING

This section outlines the concepts behind the Social Filtering formalism. We limited ourselves to the definitions we used in the RSs we tested. For a comprehensive description of this theoretical framework, please refer to [4].

In the RS domain, widely exploited in the marketing industry, it is usual to refer to *users* "consuming" *items*. Here we will employ the vocabulary associated to the museum context: *visitors* "interact" with *POIs* (any object liable to be exposed in a museum). In our experiments (see section V) we will consider POIs "liked" by visitors.

A. Bipartite Graph Visitors×POIs

The SF recommending approach is based on Social Network Analysis. More precisely, it relies on a *bipartite graph* (or *network*) and its *projections*. The bipartite graph we consider is defined over two separate set of nodes: visitors and POIs. A link can only exist between two nodes in different sets. For instance, links connect a visitor to the POIs he has viewed or liked depending on the semantic meaning we choose for the links. Such data structure can be represented by a binary *interaction* (or *preferences*) *matrix* R with L rows corresponding to the visitors and C columns corresponding to the POIs. Matrix R is thus of dimensions $L \times C$. The value r_{ui} at row u and column i is one if visitor u is connected to POI i, and zero otherwise. We denote:

- $r_{u\cdot}$ the line vector of matrix R corresponding to visitor u and $\overline{r_{u\cdot}} = \frac{1}{C}\sum_{i=1}^{C} r_{ui}$ the average number of POIs liked by u;
- $r_{\cdot i}$ the column vector of matrix R corresponding to POI i and $\overline{r_{\cdot i}} = \frac{1}{L}\sum_{u=1}^{L} r_{ui}$ the average number of visitors who liked i.

B. Graph Projections

The bipartite graph is then projected into two (unipartite) graphs, one for each set of nodes: a Visitors Graph and a POIs

 ISBN: 978-1-61208-415-2

Graph. In the projections (see Figure 1 for a toy example), two nodes are connected if they had common neighbors in the bipartite graph. The link weight can be used to indicate the number of shared neighbors. For example, two visitors are connected if they have liked at least one same POI (we usually impose a more stringent condition: at least K POIs). The projected networks can thus be viewed as the network of visitors that liked at least K same POIs (visitors having the same preferences) and the network of POIs liked by at least K' same visitors. Thus, such projected networks are *implicit social networks*: they do not follow from a deliberate social connection like in usual explicit social networks (e.g., friendship networks on Facebook). Instead, they reflect relations derived from similar behaviors of the visitors.

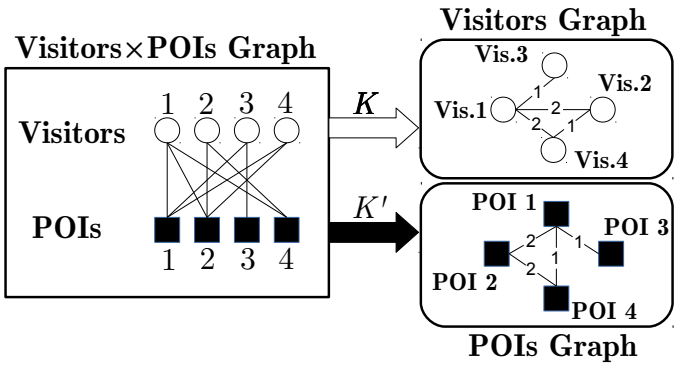


Figure 1. Bipartite Visitors×POIs graph and its projections (K=K'=1).

The general idea of Social Filtering (SF) is to leverage the concepts and methods of network analysis by exploiting the central hypothesis of social recommendation: connected entities (visitors or POIs) are *similar* in some way and thus share tastes or attributes. This property is known as *homophily* [5].

Network structures allow to define similarity between instances, neighborhoods and communities that can be relevant, as we will see, to suggest content to museum visitors. We apply these techniques to the visitors (user-based recommendation) or the POIs projected graphs (item-based recommendation).

C. Similarity Measures

We consider an *active visitor* a for whom we seek recommendations, u being any other visitor. *Asymmetric cosine similarity* [6] is a flexible way for defining the similarity between them:

$$\text{Sim}_{\text{asymcos}}(a,u) = \frac{r_{a\cdot} \cdot r_{u\cdot}}{\|r_{a\cdot}\|^{2\alpha}\|r_{u\cdot}\|^{2(1-\alpha)}} \quad (1)$$

where $r_{a\cdot} \cdot r_{u\cdot} = \sum_{i=1}^{C} r_{ai} r_{ui}$ denotes the dot product of vectors $r_{a\cdot}$ and $r_{u\cdot}$, $\|\cdot\|$ is the associated euclidian norm and α is a real number in $[0,1]$. Note that for $\alpha = 0.5$ we obtain the classic cosine similarity.

The similarity between two POIs i and j (not displayed here for space-saving purposes) can be defined in the same fashion simply by replacing visitors by POIs or, equivalently, rows by columns.

Many more similarity measures are implemented in the SF framework that are not described here because there were not used in the experiments.

D. Neighborhoods

Given a network and a similarity measure we can now define the neighborhood $K(a)$ for an active visitor a (we only give the definition for visitors; neighborhood $V(i)$ for a POI i is defined in a similar manner):

- $K(a)$ is the first circle of neighbors of a in the Visitors Graph, where they can be rank-ordered by their similarity to a.
- $K(a)$ is the *local community* of a in the Visitors Graph, where local communities are defined as in [7].
- $K(a)$ is the *community* of a in the Visitors Graph where communities are defined, for example, by maximizing modularity [8].

As stated in [4], these last two cases are novel ways to define neighborhoods for recommendation systems: visitors in $K(a)$ might not be directly connected to the active visitor a. These definitions thus embody some notion of paths linking visitors through common behavior patterns.

E. Scoring Functions

The last step in the RS pipeline consists in providing a ranked list of recommended POIs to the active visitor a. This is done through a *scoring function* that aggregates the preferences concerning a given POI i. The user-based approach considers the preferences of the neighbors in $K(a)$ about i:

$$\text{Score}_{\text{U}}(a,i) = \sum_{u \in K(a)} f\big(\text{Sim}(a,u)\big)\ r_{ui} \quad (2)$$

Alternatively, the item-based approach takes into account the preferences of a on POIs in the neighborhood $V(i)$ of i:

$$\text{Score}_{\text{I}}(a,i) = \sum_{j \in V(i)} r_{aj}\ g\big(\text{Sim}(i,j)\big) \quad (3)$$

Various functions f and g can be used [9]. For the sake of brevity, we only mention the scoring functions we applied (alternatives are thoroughly described in the reference paper on SF [4]):

- *weighted average popularity* for a of POI neighbors of i in $V(i)$ weighted by their similarity to i:

$$\text{Score}_{\text{I}}(a,i) = \frac{1}{\sum_{j \in V(i) \cap I(a)} |\text{Sim}(i,j)|} \sum_{j \in V(i)} r_{aj}\, \text{Sim}(i,j) \quad (4)$$

  where $I(a)$ is the set of POIs liked by a.
- *scoring function "with locality"*: Aiolli [6] proposed another mechanism to produce locality without having to explicitly define neighborhoods. Function g is defined so as to put more emphasis on high similarities (with high q'):

$$\text{Score}_{\text{I}}(a,i) = \sum_{j \in V(i)} r_{aj}\ \big(\text{Sim}(i,j)\big)^{q'} \quad (5)$$

Finally, POIs are rank-ordered by decreasing scores and the top k POIs $(i_1^a, i_2^a, ..., i_k^a)$ are recommended to a, where $\text{Score}(a, i_1^a) \geq \text{Score}(a, i_2^a) \geq ... \geq \text{Score}(a, i_k^a)$

 ISBN: 978-1-61208-415-2

III. Classical Recommender Systems

Many recommendation techniques are commonly used in industry. We now briefly describe those we used as baseline or for comparison purposes. Some of these methods can be expressed as special cases of the SF formalism.

A. Popularity

Perhaps the simplest recommendation method: we rank POIs by decreasing popularity (the sum of ones in each column of matrix R) and suggest the list of top k most popular POIs to the active visitor. This method is used as a baseline.

B. Collaborative Filtering (CF)

CF is a widely used technique to implement RSs. There exist two main groups of CF techniques: *memory-based* (or neighborhood methods) [9] and *model-based* (or latent factor models) [10]. CF methods use the opinion of a group of similar visitors to recommend POIs to the active visitor.

1) Memory-based Methods: as SF, these techniques rely on the notion of similarity between visitors or POIs to build neighborhood. Unlike content-based methods, that we did not implement, similarity is not computed on the basis of the attributes of the instances (visitors or POIs). Instead, it is based on the shared preferences between two visitors (user-based CF) or the number of common visitors who liked two given POIs (item-based CF). The ways for computing the similarity are the same as described in the Section II-C. In fact, it is easy to observe that CF can be obtained with the SF framework by choosing $K = K' = 1$ as parameters of the projected graphs, cosine similarity (eq. 1 with $\alpha = 0.5$) and a score function as in Section II-E.

2) Model-based Methods: Model-based RSs estimate a global model, through machine-learning techniques, to predict ratings. This generally leads to models that neatly fit data and therefore to good quality RSs. However, learning a model may require a great amount of training data which could be a problem in some applications. Many model-based CF systems have been proposed [11]. One of the most efficient and used model-based methods is *matrix factorization* [12] in which visitors and POIs are represented in a low-dimensional latent factors space. This technique is more suited to feedback with ratings (e.g., zero to five "stars").

C. Association Rules

Association rules mining [13] is a popular technique widely used in marketing in order to find regularities in large databases like products often purchased together. Association rules of length two can be used for recommendation [14]. They are equivalent to the item-based SF choosing asymmetric confidence-based similarity with the suitable parameters, but we preferred to use the classic Apriori algorithm [15] to implement this method.

IV. Evaluation of Recommender Systems

This section recalls the evaluation methods listed in the corresponding part of [4].

For a given active visitor, a RS produces a list of ranked POIs. We want to evaluate whether they are adequate for him. Two scenarios may be considered to evaluate RSs:

- *online evaluation*: if live interactions between visitors and POIs are available we can build RSs on past behaviors and measure the reaction of visitors to the suggestions: does the visitor take them into consideration, like them, etc.? Several groups of control can be considered in order to test different recommendation strategies. This approach is used by merchant websites, for example.
- *offline evaluation* relies on a static dataset of interactions between visitors and POIs on which we simulate recommendation. We underline the fact that this dataset corresponds to visits without recommendation. As it is usual in the evaluation of machine-learning algorithms, the original dataset is split into a training set and a test set. For each visitor of the test set, considered as an active visitor, recommendations are computed based on the data from the training set and from part of the interactions between the visitor and the POIs, taking into account time stamps if available. Evaluation is then computed by comparing the recommended POIs with the remaining real interactions the active visitor had with the POIs not taken into consideration for the recommendation computation.

One may argue that this last approach is flawed since the active visitor would probably have behaved differently if he had been actually recommended with POIs. Moreover, one might also question the relevance of evaluating RSs on the basis of the accuracy to predict the POIs the active visitor liked without being recommended, since the recommendation principle is precisely to suggest contents that the visitor would not have been likely to discover without being recommended. Nevertheless, although these arguments are valid when there exists a vast choice of items like in most marketing situations, in a museum exhibition it is reasonable to assume that the visitors interacted with almost all of the available POIs. Thus, predicting his appreciation on part of the POIs is valuable to evaluate the performance of a RS. Naturally, offline evaluation is unable to take into account the influence of being recommended: there is a psychological bias that is beyond the scope of this study.

A. Performance Metrics

In both situations, for each active visitor of the test set we have a *target set* T_a that represents the set of POIs he liked after being recommended. Let $R_a = (i_1^a, i_2^a, ..., i_k^a)$ be the set of k POIs recommended to a. The metrics classically used in this context are:

- Precision@$k = \frac{1}{L} \sum_a \frac{|R_a \cap T_a|}{k}$
- Recall@$k = \frac{1}{L} \sum_a \frac{|R_a \cap T_a|}{|T_a|}$
- Mean Average Precision:
  MAP@$k = \frac{1}{L} \sum_{a=1}^{L} \frac{1}{k} \sum_{i=1}^{k} \frac{C_{ai}}{i} 1_{ai}$

where C_{ai} is the number of correct recommendations to visitor a in the first i recommendations (Precision@i for visitor a) and $1_{ai} = 1$ if POI at rank i is correct (for visitor a), 0 otherwise.

B. Qualitative indicators

Additionally, more "qualitative" metrics indicate whether all visitors (resp. POIs) receives recommendations (resp. are recommended) or which of the more or less popular POIs are recommended: as we will see, some RSs might be better on

 ISBN: 978-1-61208-415-2

performance metrics and poorer on these qualitative indicators. Let U_{test} denote the set of visitors in the test set and $L_{\text{test}} = |U_{test}|$ the number of visitors in it, then:

- VisitorsCoverage@$k = \frac{\text{nb visitors in } U_{test} \text{ with } k \text{ reco}}{L_{\text{test}}}$

  is the proportion of visitors who get recommendations.

- Average number of recommendations: when visitors coverage is not 100% , i.e, not all visitors got k recommended POIs, we may want to know the average number of POIs recommended for the visitors with partial lists:

  $\text{AvNbRec@}k = \sum_{K=0}^{k-1} K \frac{\text{nb visitors in } U_{test} \text{ with } K \text{ reco}}{L_{\text{test}} - \text{nb visitors in } U_{test} \text{ with } k \text{ reco}}$

- POIs coverage: a high diversity of suggested POIs should result in more attractive recommendations. We thus seek a high proportion of POIs that are recommended:

  $\text{POIsCoverage@}k = \frac{\text{nb distinct POIs in reco lists}}{C}$

- Head/Tail coverage: if we rank POIs by decreasing popularity (number of visitors who liked each POI), we call *Head* the 20% of POIs with highest popularity and *Tail* the remaining 80%. Recommending only most popular POIs will result in relatively poor performances and low diversity. We thus define the rate of recommended POIs in the Head and in the Tail:

  $\text{RateHead@}k = \frac{1}{L_{\text{test}}} \sum_{u \in U_{test}} \frac{\text{nb reco for } u \text{ in Head}}{\text{nb reco for } u}$

  $\text{RateTail@}k = 1 - \text{RateHead@}k$

C. Accuracy vs.Originality

Ideally we would like to produce accurate recommendations that are not too popular, providing the visitors with "pleasant discoveries". This amounts to maximize both MAP@k and RateTail@k. Furthermore, it could be interesting to give more or less emphasis to each of these two metrics depending on the objectives of the recommendation: accuracy vs. originality (or novelty). We propose the following (not normalized) combined indicator:

$$\text{Perf}_e = e\ \text{MAP@}k + (1-e)\ \text{RateTail@}k \qquad (6)$$

where e is chosen in $[0, 1]$ depending on the relative importance we want to give to each aspect of the performance.

Now that we have described how we build RSs and evaluate their performances it is time to expose our experimental results on a dataset extracted from a real museum exhibition.

V. DATASET

This section describes the origin and principal features of the dataset we used to experiment on RSs for museums.

A. General description

From March 11 to August 24, 2014, the Great Black Music exhibition (GBM) took place at the Cité de la Musique in Paris [16]. Both Cité de la Musique and the curator M. Benaïche (director of the digital art factory l'Atelier 144) are members of the AMMICO project consortium. The exhibition showcased the variety and story of the black music around the world by means of numerous multimedia installations. It has been successful with around 76 000 unique visitors.

At the entrance, the visitor got a stereo headset connected to a "smartguide" device which was an Android smartphone running a specifically developed application. Several technological solutions are explored nowadays in order to have direct and accurate information about which exhibition items is viewed by a specific visitor. In the GBM exhibition, visitors simply introduced manually the POI identification number displayed on it in the exhibition space. Since a significant amount of the content was only available through the device (e.g. musical content), visitors were highly motivated to use it. With this equipment, the visitor was able to interact with numerous audiovisual material (11 hours of available recordings in total). Among other features, the device allowed him to create his personal playlist by "liking" (or bookmarking) his favorite contents (POIs) that he could later retrieve online by logging into a dedicated personal webpage [17]. This possibility was a fairly good incentive for visitors to bookmark POIs. On the example displayed on Figure 2 the visitor can add POI n°23 (artist: Tumi & The Volume; song: Asinamali) on his favorites playlist. .

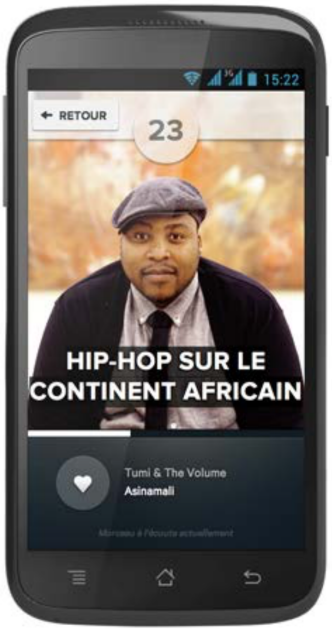


Figure 2. Example of the user interface used at GBM exhibition.

On the museum side, this setting allowed to collect a large amount of data on visitors' behaviors: each time they interacted with a content by the means of the device, which was indispensable given the very nature of the exhibition, the details of the action (visitorId, POIId, time, duration, liked or not) was recorded in the exhibition database. In total, more than 20 million interactions were recorded concerning all the 75 774 visitors.

From this raw database we constructed a dataset focusing on the bookmarked POIs: we considered the bipartite graph consisting of the two sets U and I of the visitors and the liked POIs respectively, where a link connecting a visitor u with a POI i means that "u liked i". We ended up with $|U| = 67\,883$ users, $|I| = 600$ liked POIs (among 608 possible POIs to bookmark) and $|E| = 1\,681\,534$ links (bookmark notifications) between the two sets. Visibly, around 10% of the visitors ($75\,774 - 67\,883$) did not make use of the bookmarking functionality. For them, other strategies might be implemented

TABLE I. GBM DATASET STATISTICS

number of visitors $\|U\|$	67 883
number of POIs $\|I\|$	600
number of "likes" $\|E\|$	1 681 534
min and max visitor degree	1...447
mean and std visitor degree	24.8 ± 34.4
min and max POI degree	1...13 005
mean and std POI degree	2802.6 ± 2481.4

in order to provide recommendation, for example taking into account the time they spent viewing a POI as a measure of their interest. Since this study aims at evaluating different recommendation methods and not at providing explicit live recommendation to visitors we simply excluded them from the dataset.

B. Degree distribution

A vast majority of visitors bookmarked a relatively small amount of POIs and as the number of bookmarked items per visitor increases less visitors are concerned. We thus get a typical *power-law* distribution of visitors nodes' degrees.

POIs are also unevenly popular: 19 of them received a single "like" from the visitors, while others were bookmarked by a large amount of them. The three top favorite POIs are the ones corresponding to the songs "Why I Sing the Blues" by B.B. King, "Sodade" by Cesaria Evora and "Respect" by Aretha Franklin, liked respectively by 13 005 visitors (19.2% of the visitors and 0.77% of the "likes"), 11 801 visitors (17.4% of the visitors, 0.70% of the "likes") and 11 552 visitors (17.0% of the visitors, 0.69% of the "likes").

C. Mega-hubs

In a network, *hubs* are nodes with the highest degree. They are common in social networks as a consequence of the power-law degree distribution. *Mega-hubs* are nodes connected to all or almost all the other nodes of the graph. We generally consider them as not informative in the recommendation context. Moreover, they could undermine the performance of RSs by not allowing the meaningful communities to emerge. Also, from a technical point of view, the presence of mega-hubs causes increased loads of computation and memory. These considerations often lead practitioners to remove mega-hubs from the networks.

What about our dataset? The highest POI degree in the bipartite graph is 13 005 out of a maximum of 67 883 possible links to visitor nodes. The corresponding node in the POIs Graph is a hub connected to less than 20% of the nodes. With respect to visitors, the highest degree is 447 out of a maximum of 600. This relative high value ($\approx 75\%$ of the possible links) indicates the presence of potential mega-hubs in the Visitors Graph. There are several ways to define mega-hubs, but we will not enter into details here: in this work-in-progress study we first used the entire original dataset without eliminating the potential mega-hubs.

Table I summarizes some statistics of the GBM dataset.

VI. EXPERIMENTS

We performed an offline evaluation of several RSs on the GBM dataset. As explained before, it consists in simulating a recommendation scheme and comparing suggested POIs to some visitors with the POIs they actually liked.

A. Experimental Setting

We randomly split the data into 90% visitors for training and the remaining 10% for testing. For training, we used all the interactions (likes) of the 90% visitors. For testing we input 50% of the interactions of each test visitor and compared the obtained recommendation list to the remaining 50% liked POIs.

The RSs methods we chose to evaluate are (see Sections II and III):

- Popularity: used as a baseline;
- Bigrams: association rules of length two implemented using Apriori algorithm [15] with thresholds on support and confidence at 1%. POIs are ranked in decreasing confidence of the rule generating them;
- NMF (non-negative matrix factorization): we used the code associated to [18], with maximal rank 10 and maximum number of iterations 50. These parameters were chosen after several attempts at maximizing the performances, but without a systematic exploration of their value space.
- CF_UB: user-based collaborative filtering implemented as a special case of SF with cosine similarity (eq. 1 with $\alpha = 0.5$) and weighted average popularity (eq. 4 adapted to visitors) as scoring function;
- CF_IB: item-based collaborative filtering implemented as a special case of SF with cosine similarity (eq. 1 with $\alpha = 0.5$) and weighted average popularity (eq. 4) as scoring function;
- SF_IB: item-based social filtering with asymmetric cosine similarity (eq. 1). The neighborhood is defined as the top 10 most similar neighbors in the first circle of neighbors of the POIs graph and we used the scoring function with locality (eq. 5). We explored several combinations for the parameters α (of similarity) and q' (of the scoring function) and reported the most interesting results.
- We tried user-based SF with different parameters α and q, but it shed poor results that we will not report here.

We produced suggested POI lists of length $k = 10$ and evaluated the RSs using all the indicators described in Section IV. In order to obtain more accurate measures we repeated the process on 30 different randomly split training/test sets (90%-10%) and computed the mean value for each indicator.

B. Results

Members of the L2TI laboratory implemented the SF formalism in a Python library released under an open source license [19]. A flexible processing pipeline and the versatility of our SF formalism provided an efficient way to assemble the various elements for experimenting on several methods.

The performances are shown in Table II. Values in **bold** and *italic* indicate respectively the best and second-best performances for the corresponding indicator (except for computation time, "higher is better" for all the performance indicators). We ran our simulations on an Intel Xeon E7-4850 2,00 GHz (10 cores, 512 GB RAM), shared with members of the team so that concurrent usage may have happened in some of the experiments, with impact on reported time. Computing time

 ISBN: 978-1-61208-415-2

TABLE II. PERFORMANCES OF RECOMMENDATION SYSTEMS ON GBM DATASET.

	Popularity	Bigrams	NMF	CF_UB	CF_IB	SF_IB $\alpha = 0.1$ $q' = 3$	SF_IB $\alpha = 0.1$ $q' = 2$	SF_IB $\alpha = 0.9$ $q' = 1$	SF_IB $\alpha = 1$ $q' = 3$
MAP@10	0.035	0.080	0.004	0.005	0.077	**0.087**	*0.086*	0.049	0.015
Precision@10	0.059	0.124	0.078	0.018	0.119	**0.132**	*0.131*	0.092	0.036
Recall@10	0.080	**0.158**	0.105	0.023	*0.152*	**0.158**	**0.158**	0.113	0.067
VisitorsCoverage@10	62.60%	**100%**	*96.93%*	47.22%	85.44%	74.00%	74.23%	84.77%	83.77%
AvNbRec@10	8.52	-	4.67	3.26	6.46	5.63	5.62	5.24	4.88
POIsCoverage@10	1.69%	71.81%	20.41%	93.96%	**99.17%**	93.88%	*95.63%*	**99.17%**	94.61%
RateTail@10	0%	24.12%	6.35%	35.23%	44.83%	35.09%	36.92%	*73.64%*	**91.87%**
$\text{Perf}_{0.0}$	0.000	0.241	0.063	0.352	0.448	0.351	0.369	*0.736*	**0.919**
$\text{Perf}_{0.1}$	0.003	0.225	0.058	0.317	0.411	0.325	0.341	*0.668*	**0.828**
$\text{Perf}_{0.5}$	0.017	0.161	0.034	0.178	0.263	0.219	0.228	*0.393*	**0.467**
$\text{Perf}_{0.9}$	0.031	0.096	0.010	0.039	0.114	0.113	*0.114*	**0.118**	0.106
$\text{Perf}_{1.0}$	0.035	0.080	0.004	0.005	0.077	**0.087**	*0.086*	0.049	0.015
Computation time	**0:00:10**	0:30:00	0:30:00	*0:00:30*	*0:00:30*	*0:00:30*	*0:00:30*	*0:00:30*	*0:00:30*

is thus indicative only (0:00:10 is 10 seconds, 0:30:00 is 30 minutes).

C. Discussion

The first observation one might be inclined to make is that the performance measurements of MAP, Precision and Recall seem to be low in relation to their possible values in $[0, 1]$. However, compared to similar experiments on other datasets, we note that these apparently low values are common in the RS evaluation context (see the results on four publicly available datasets presented in [4]).

Supporting the observations reported in [6] where the author carried out the same kind of experiments but with a different dataset, the results for SF_IB show that asymmetric cosine similarity and the scoring function with locality bring enhanced performances to classic methods, provided a suitable choice for the parameters α and q'. It outperforms in all indicators except for VisitorCoverage@10: it is able to provide a full list of ten recommended POIs for a maximum of around 85% of the visitors. The remaining visitors received an average of five suggested items. It gives a significant improvement on traditional item-based CF (CF_IB) from which it is derived.

Within the variants of SF_IB when changing the parameters, we note the necessary trade-off between accuracy and originality of the recommendation: when performance metrics increase (MAP, Precision and Recall) it is at the expense of qualitative indicators, especially RateTail. This is well captured by our combined indicator Perf_e.

Following SF_IB, Bigrams presents fairly good relative performances, particularly on VisitorCoverage which is 100%. However, it has low RateTail and the computation time is 60 times longer.

User-based CF does not give good results on this dataset, similarly to all the user-based methods we tried (SF_UB). This could be caused by the presence of mega-hubs in the Visitors Graph. This point would be worth exploring.

NMF performs particularly bad on this dataset, only slightly better than the baseline Popularity. This may be due to the insufficient amount of data necessary to build an accurate model and on the fact that we consider simple binary feedback (liked or not) instead of an explicit rating with which this method is known to perform better.

Finally, the baseline Popularity behaves as expected: by recommending the 10 most popular POIs without taking into account the similarities of visitors' behaviors, its POIsCoverage is dramatically low and the RateTail is null, by definition.

VII. CONCLUSION

We have presented in this paper an evaluation study of several recommender systems applied to the museum visit context. We compared and discussed the performances of different recommendation strategies by evaluating them on a genuine dataset concerning visitor behaviors in a real exhibition. Beside classic recommendation methods, we used a versatile Social Filtering formalism developed and implemented in our laboratory. The results show that promising improvements can be achieved with efficient algorithms provided that parameters are properly adjusted.

We are currently conducting experiments regarding the neighborhoods we consider in the projected graphs: better recommendations could be obtained by taking into account the graph community or local community of the active user as described in Section II-D instead of the simple first circle of neighbors since interesting suggestions of POIs could come from related but not directly connected visitors. In parallel, we are carrying out a set of experiments on modified versions of the present dataset in order to observe the influence of mega-hubs on the recommendation quality. Combining several recommendation methods in what are commonly denominated *ensemble methods* in statistical learning is another direction that could somehow enhance performances.

This application domain raises other issues that may be interesting to investigate: how is recommending content perceived and accepted by museum visitors? Beyond the quality and relevance of the suggested content, what is the influence of the presentation and editorialization in its receptivity?

ACKNOWLEDGMENT

This work was supported by the French AMMICO project funded by Banque Publique d'Investissement (BPI) in the FUI 13 program.

REFERENCES

[1] http://ammico.fr.

[2] R. Fournier, E. Viennet, S. Sean, F. Soulié-Fogelman, and M. Bénaïche, "Ammico: social recommendations for museums," in Proceedings of Digital Intelligence (DI2014), Nantes, France, 2014.

[3] L. Ardissono, T. Kuflik, and D. Petrelli, "Personalization in cultural heritage: the road travelled and the one ahead," User modeling and user-adapted interaction, vol. 22, no. 1-2, 2012, pp. 73–99.

 ISBN: 978-1-61208-415-2

[4] D. Bernardes, M. Diaby, R. Fournier, F. Fogelman-Soulié, and E. Viennet, "A social formalism & survey for recommender systems," SIGKDD Explorations, vol. 16, no. 2, Dec. 2014.

[5] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," Annual review of sociology, 2001, pp. 415–444.

[6] F. Aiolli, "Efficient top-n recommendation for very large scale binary rated datasets," in Proceedings of the 7th ACM Conference on Recommender Systems, ser. RecSys '13. New York, NY, USA: ACM, 2013, pp. 273–280. [Online]. Available: http://doi.acm.org/10.1145/2507157.2507189

[7] B. Ngonmang, M. Tchuente, and E. Viennet, "Local community identification in social networks," Parallel Processing Letters, vol. 22, no. 01, 2012.

[8] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," Journal of Statistical Mechanics: Theory and Experiment, vol. 2008, no. 10, 2008, p. P10008.

[9] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," Knowledge and Data Engineering, IEEE Transactions on, vol. 17, no. 6, 2005, pp. 734–749.

[10] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2002, pp. 61–70.

[11] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, Recommender systems: an introduction. Cambridge University Press, 2010.

[12] P. Victor, M. De Cock, and C. Cornelis, "Trust and recommendations," in Recommender systems handbook. Springer, 2011, pp. 645–675.

[13] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," in ACM SIGMOD Record, vol. 22, no. 2. ACM, 1993, pp. 207–216.

[14] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in The adaptive web. Springer, 2007, pp. 325–341.

[15] C. Borgelt, "Frequent item set mining," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 2, no. 6, 2012, pp. 437–456.

[16] http://www.greatblackmusic.fr.

[17] http://www.greatblackmusic.fr/fr/mon-expo.

[18] J. Kim and H. Park, "Fast nonnegative matrix factorization: An active-set-like method and comparisons," SIAM Journal on Scientific Computing, vol. 33, no. 6, 2011, pp. 3261–3281.

[19] https://bitbucket.org/danielbernardes/socialfiltering.

 ISBN: 978-1-61208-415-2

An Extensible Conceptual Model for Tabular Scientific Datasets

Javad Chamanara*, Michael Owonibi*, Alsayed Algergawy*, and Roman Gerlach†

Friedrich Schiller University of Jena
*Institute for Computer Science
†Institute for Geography
Jena, Germany
Email: firstname.lastname@uni-jena.de

***Abstract*—There is a proliferation of datasets generated by various scientists of different scientific disciplines. Therefore, there is a growing need to construct and develop platforms that enable scientists to capture, exchange, process, and interpret data for immediate use, as well as to store and manage data to support future reuse. Modeling and organizing data within such platforms are key challenges. To this end, in this paper, we introduce the dataset model of the *BExIS 2* platform and how data can be organized inside the model. In particular, we describe the anatomy of a general purpose tabular dataset, which consists of data tuples to represent the table rows and data cells that are compound objects holding the obtained values and their auxiliary information. The structure of datasets is defined and applied separately in order to factor out shared concepts such as unit of measurement, methodology, data type, valid and missing values, processing functions and so on. The datasets are extensible in multiple ways and can be annotated on various levels utilizing taxonomies, ontologies, and custom metadata structures.**

Keywords–Scientific data; Dataset structure; Biodiversity data.

I. INTRODUCTION

In research data management, one of the utmost goals is to support data sharing, as this facilitates the reproduction and evaluation of scientific results as well as the reuse of the data for other purposes. Traditionally, researchers focused on collecting, processing and analyzing data and then published their findings in the scientific literature. Preparing and publishing research data was not part of the general scientific workflow. This has been changing. Publishing data is becoming a standard in most disciplines thanks to the advent of dedicated data repositories (e.g., Dryad [1], Pangaea [2]), data journals (e.g., Natures Scientific Data [3], Earth Science Data Journal [4], Biodiversity Data Journal [5]) and funding organizations requesting data publication. With such publications data becomes persistently available, documented, citable, and to some extent validated [6].

Many of these data repositories follow a rather generic approach to data management and accept a broad range of data models, data formats, and data types. They provide facilities to store data as files, together with a description of the content, structure, and administrative information in metadata documents. For some repositories (e.g., Pangaea) data submission is a curated process, which improves data quality in terms of consistency, completeness, and reusability. But the primary focus is still to make data discoverable by humans and allow them to download data files. When thinking about reuse in the sense of (automatic) data integration additional requirements need to be satisfied, e.g., flexible access patterns (selection and projection), data change/update/provenance management, integrated analysis, human and machine interpretability, flexible security and access management, data context provisioning, and semantic enablement. For instance, it will be really difficult to automatically integrate datasets which have not been parsed in the first instance (the dump table files), and in instances where they are dynamically parsed, the question of determining equivalent variables in different datasets and unit conversion comes into play.

A study conducted by Rexer [7] has shown that more than 90% of datasets contain less than 100 million records and are mostly managed/ processed by tools such as RDBMSs, Excel, or R. Another study done recently by O'Reilly indicated tabular datasets are among the most used forms of data [8]. This is due to the popularity in usage of spreadsheets for handling (storing and analyzing) data by the data providers, which is in turn due to the fact that spreadsheets are relatively easy to use, flexible, and compatible with a lot of applications across several disciplines. Also many of data acquisition tools simply generate raw data in tabular form, mostly comma separated flat files.

In this paper we focus on the domain of biodiversity, where spreadsheets, relational databases and statistical tools like R are widely used for managing data [9]. Biodiversity data is highly heterogeneous, including information about species distribution and abundance, genetic sequences, trait measurements, organisms, their morphology and genetics, life history and habitats, and geographical ranges. These data is mostly linked to spatial, temporal, and environmental data [10][11][12]. These heterogeneities can be broadly classified into five categories: technical, syntactic, structural, semantic, and data models [13]. Data model heterogeneity is the problem that systems and tools employ different data models, such as relational, XML, or semantic-based data models. A recent study shows that most existing biodiversity repositories are based on relational database models [12]. In contrast, structural heterogeneity focuses on the problem that information can be represented in multiple ways for a given data model.

Therefore, in order to effectively manage tabular data in a data repository, there is a need to model the composition of tabular datasets such that it satisfies the manifold data management needs outlined above. The current paper is an effort to extend the conceptual model presented in [14] and provide more details on the concept of a generic dataset. Although the model was developed for this particular domain we expect it to be applicable to others as well.

 ISBN: 978-1-61208-415-2

The rest of the paper is organized as follows: a brief survey of related work is presented in the following section. We introduce our proposed data model in Section III and then elaborate its flexibility and extensions in Section IV. Finally, we conclude the paper and outline future work in Section V.

II. Related work

Recently, the World Wide Web Consortium (W3C) attempted to standardize the description of tabular data [15], such that the tabular data is structured into rows, each of which contains information about some thing. Each row contains the same number of cells providing values of properties of the thing described by the row. The W3C initiative broadly classifies tabular data into three main models: a *simple table*; consisting of columns, rows and cells with no form of annotation, an *annotation table*, i.e., a table annotated with additional metadata, and a *group of tables* comprising of a set of tables and a set of annotations that relate to the dataset.

Similarly, the INSPIRE Observation and Measurements standards (O&M) aims to normalize the representation of records of scientific measurement [16]. It introduces the notion of observation as an event whose result is an estimation of the value of some property(ies) of a feature-of-interest, obtained using a specified procedure. O&M defines a core set of properties for an observation, and these include the feature of interest, observed property, result value, procedure (the instrument, algorithm or process used), event specific parameters (e.g. instrument setting), phenomenon time, etc. One physical realization of this model is the tabular data. Users of this standard are not only able to describe features and properties but also to organize and store data. While the O&M standard was developed in the context of geographic information systems, the model is not limited to spatial information.

The Statistical Data and Metadata Exchange (SDMX) initiative [17] also sets standards that can describe and facilitate the exchange of statistical data and metadata. Based on the standard, every dataset will have a data structure definition, which specifies the organization of a data set. In addition, each column in the table can either be a function as a dimension, a measure, or an attribute. They may also play a role based on a set of roles defined in the standard e.g. identity, time format, frequency. Every column in the table is also based on a concept which has to be defined before the creation of the column. Different organizations can implement the standard and use it to exchange datasets. For instance, many of the datasets in Eurostat are implementations of this standard. Typically, a group of data providers defines an implementation of the standard which is used within the group, e.g., Balance of Payments data exchange, National Account data exchange.

Pangaea [2] is a repository for managing tabular data. It is an information system aimed at archiving, publishing, and distributing data related to earth science fields. The challenge of managing these heterogeneous data was met through a flexible data model. In this model, a dataset is modelled as a collection of data series and a data series consists of one or several data points for one parameter (table column). Information about the parameters, e.g., parameter unit and collection method is documented. This information can be used to parse, store and read the actual tabular data, which is stored independently of its description.

It is clear that tabular data has become widely used not only in generic domains but also in scientific data. One of these domains is biodiversity data. As a consequence, a number of repositories have been developed. In the following, we present some of these repositories, focusing on how they model and organize data. BEFdata [18] is a software platform providing support for interdisciplinary data sharing and harmonization for collaborative research projects [19]. It provides functionalities for the upload, validation, and storage of data from a formatted Excel workbook. A collection of columns (variables) in the main Excel sheet then establishes a dataset. During data upload, the Excel sheet containing the main tabular data is decomposed into its sheet-cells at the database level so that each and every single primary data value is stored independently in a database table row. Each value is thus uniquely identified in this integrating table by its source table identifier, its source table variable identifier, and its source table row identifier.

In addition, other repositories exists e.g., the Biodiversity Exploratories BExIS (BE BExIS) [20], BCO-DMO [21] that archive tabular data either as dump of the original files or in some relational forms, and provide some functionality for describing the structure of the tabular data [11][12]. In BE BExIS, tabular data (referred to as primary data) is a collection of *"observation"* entities so that each observation record is a set of values related to a specific observation. The data structure introduces the list of variables, so that each variable at least has a name, data type and a description. These information are stored as part of the metadata of the dataset. BExIS keeps track of all editing and deletions of the observations of datasets by means of a versioning mechanism.

One further direction with reference to modeling tabular data is to semantically enhance the data by using different methods, such as taxonomies [22], metadata, and ontologies [12]. For example, a wide range of metadata standards have been established over the last decade, such as EML [23] for ecological data and ABCD [24] for collection data. Ontologies can be viewed as extensions of metadata standards and are the most fundamental approach to address the problem of semantic heterogeneity. The goal of an ontology is to describe not only data, but the knowledge behind the data. One quarter of the existing repositories for biodiversity data uses ontologies, such as OBOE (Extensible Observation Ontology), as a flexible solution for standardizing attributes and their relationships [10][13][25][26].

III. Core Dataset Model

The current work is a continuation of [14], which presents a general purpose conceptual model for scientific data management. A dataset, in the model, plays the role of a data container for observations, measurements, simulations, and other supported forms of data. The meaning of data is determined by its bound data structure, which in turn determines the columns of the dataset by introducing the variables. The variables define among others the name, data type, unit of measurement, methodology and procedure of obtaining data, and measurement scale. The reusable elements of variables such as units of measurements, unit conversion information, data types, and data validation rules are factored out into *Data Container* concepts, to make data sharing, integration, and cross querying easier.

 ISBN: 978-1-61208-415-2

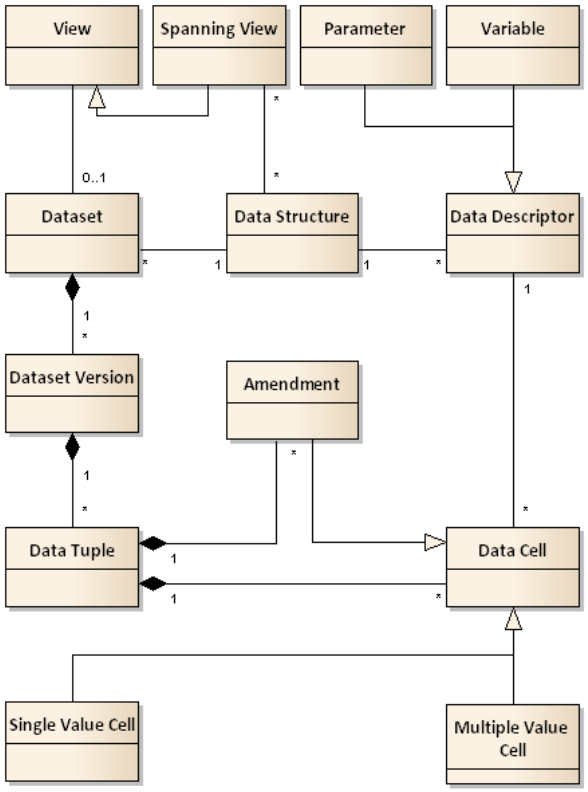


Figure 1. Conceptual model

In this paper, we look at the internals of the dataset and explain its elements in more detail. A *Dataset*, in our design, is a set of, possibly duplicate, *Tuples*. Each tuple is a collection of *Data Cells* containing the *Data Items* as shown in Fig. 1. Each cell is a compound data structure able to hold single or multiple values resulting from observations, measurements, computations, simulations, or any other means of data acquisition. In addition to the data values, the cells contain sampling, result times, and descriptions about the values, and most importantly the link to their formal description, which is captured by the concept *Data Descriptor*. The reason why sampling and result time are captured separately is that in physical object samplings, the sample may have been taken in a time different than the measurement or observation. This time difference is a considerable factor for some analyses, e.g., in soil sample water or gas containment.

As the model in Fig. 1 shows, each data cell should be associated with its corresponding variable or parameter. The variables and parameters are generalized under the the *Data Descriptor* concept, as they share almost all of their attributes. The only difference between them is that the parameters are considered to be auxiliary data to a variable. An example of such an auxiliary data would be the GPS location of a tree, whose diameter at breast height is measured. Data descriptors act as table headers to determine the name, data type, unit of measurement, methodology, and other important attributes of the columns of datasets. Factoring out the variables, units, and data types not only encourages reuse, but also establishes a foundation for data harmonization, integration, and discovery. For example, an analysis process that needs data from multiple datasets may merge the relevant columns by converting their units of measurement to a single consistent one, or searching for datasets containing temperature variables having values above 20 degree Celsius may also return datasets containing temperature values greater than 68 degree Fahrenheit. More sophisticated dataset integrations can be powered by annotating the variables with ontologies and applying semantic matching algorithms to find equivalent columns among datasets.

IV. Dataset Model Extensions

The base model is capable of materializing a table, but it may not be enough for some special requirements. In addition to the basic tabular form of the datasets, the following extensions are available to all datasets.

Amendments are special kinds of data cells scientists can attach to specific tuples, as shown in Fig. 2 as *Amendment Class* inherited from *Data Cell* and associated with *Data Tuple*. Like a usual data cell, they have their own data descriptor linked to them, hence all other attributes like unit of measurement, methodology, measurement scale, and so on. Different tuples may have different numbers of amendments each linking to their designated data descriptor. Capturing exceptional observations would be an example of using amendments. There is no need for all the tuples to have the same set of amendments. Also there is no need for the amendments of various tuples to be associated to the same variables.

Although we have tried to enrich the data descriptor class with as many attributes as possible, there are cases where scientists need more data about the variables. For example, if the values of a column are obtained using a special model of a sensor, which has a known exceptional error margin, the scientist may be interested in capturing the sensor model or the error margin as a property of the column, to use it in the analyses to be done on the column. Also the measurement system calibration, configuration, and environmental parameters are proper candidates to be modeled using extended properties. These kinds of information are column level in the scope of the dataset that contains data. Fig. 2 shows an example of this extension by attaching error, rounding indicator, and resolution properties respectively to the Soil_N, Tmp (temperature), and Time variables. To summarize it, an *Extended Property* is a user defined, dataset specific attribute whose value applies to a single column.

Sometimes, the scientists need to reduce the size of a dataset by means of removing some of the columns or filtering out the data tuples in order to perform a fast experimental analysis on the data. *Views* are proper tools to extract a subset of datasets namely for processing, sharing, or sampling purposes. Also the views can be used for security or digital right management, so that a small insensitive portion of data is exposed to the public and the original dataset is kept secure. The views can filter both the visible columns and data tuples. A view applies to a single dataset, but a *Spanning View* applies to multiple datasets that use the same structure. View 1 shown in Fig. 3, has filtered all the variables of Fig. 2's sample dataset, except the Soil_Moi., Depth, and Hu variables, as well as the data tuples matching the $Depth < -10$ predicate. View 2 in the same figure has only hidden the Depth, Pos., Hu., and Temp variables.

 ISBN: 978-1-61208-415-2

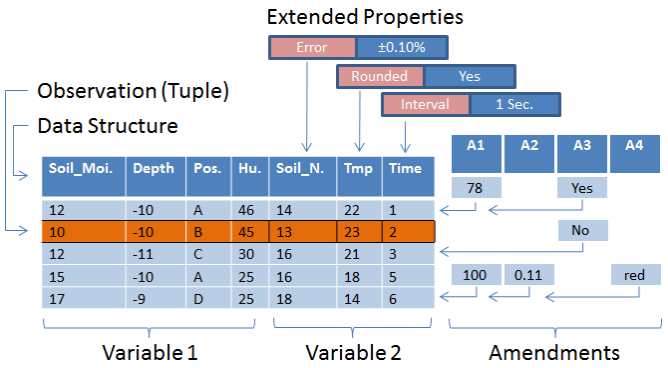


Figure 2. A sample dataset with its elements described. Soil_Moi. is Soil Moisture, Pos. is Plot code, Hu. is Humidity, Soil_N is Soil Nitrogen, and Tmp is the Temperature..

View 1 (filtered)

Soil_Moi.	Depth	Hu.
12	-10	46
10	-10	45
15	-10	25
17	-9	25

View 2 (spanning)

Soil_N.	Soil_Moi.	Time
14	12	1
13	10	2
16	12	3
16	15	5
18	17	6

Figure 3. Two exemplary views. View 1 filters some of the variables and tuples. View 2 filters some of the datasets variables.

A small but useful customization feature of the model is its multi-lingual support for variable names. It helps multi-language teams working on the same dataset have their native names on the columns. This feature potentially reduces the effect of information loss and naming inaccuracy caused by translating the domain terminologies.

As shown in Fig. 1, each dataset can have multiple versions. The data tuples belong to the versions. This versioning scheme helps to freeze the versions so that they are accessible for later processing and citations, independent of the following changes. Technical details of the versioning are not in the scope of this paper, but as a short description, it provides a check-in, check-out mechanism, computes and stores the difference between the versions. The information collected in the core and extended mode entities can serve an additional role of being treated as metadata. For example, a dataset export tool can, in addition to the actual data, extract some parts of the variables as metadata and serialize them alongside with. The datasets have metadata at three levels. Cell level metadata captures how the value was obtained, when it was sampled if so, when the result was ready, and a free-text description. Structural level metadata are handled by defining the variables, parameters and extended properties under a data structure. The dataset version level metadata are captured by user-defined or standard metadata schemas e.g., EML or ABCD. The version level metadata is describing the whole dataset version as a unit of data and may consist of various aspects among them authorship, geographical extent, copyrights, sensors or measurement tools, or software configuration. Each version of a dataset may have its own metadata, so that changes in the metadata are aligned with changes in the data.

In addition to the mentioned capabilities, the variables are able to be linked to semantic elements such as terminologies, taxonomies, or ontologies. This features make the model a proper candidate for automatic schema matching, data integration, multi-project joint analyses and so on.

V. Conclusion and Future Work

Biodiversity data has become more and more important, therefore, there is a growing need to develop new platforms and infrastructures that facilitate creating, storing, reusing, and sharing scientific data. To this end, in this paper, we introduced a data structure for tabular scientific data. In particular, we presented the core elements in the model, including dataset, dataset versions, tuples, and data cells as well as the possible extensions to these core elements. The model can be used to enforce the structure and type of information to be collected as well as a base for data validation. The attributes assigned to the variables, e.g., unit of measurement, semantic annotations, and the unit conversion information can be used in data integration efforts. Datasets published using this model allow the following researchers to obtain the data with its structure and the meaning of the elements, so that they can run similar analyses to validate or reproduce the original work, or use it in their own work. In addition, the dataset versions provide a strong framework for dataset citation.

The model lacks some features like user-defined data types for the cells and versioning the views. Currently, there is a predefined set of data types introduced to the model, so that all the cells, whether single or multiple value, accept data of those types only. It would be an improvement to allow the model users to define their own scalar or complex data types and use them in their dataset modeling needs. As described, the views can reduce the amount of visible data of target datasets. A useful feature of the model would be to apply the versioning concept to the views too, so that they can be attributed or cited independently guaranteeing access to the same subset of datasets over time. In our future work, we are going to extend the model to address these shortcomings.

Acknowledgments

This work was partly funded by the German Science Foundation through the projects BExIS++, AquaDiva (subproject INFRA1), and Biodiversity Exploratories (subproject Data Management).

References

[1] “Dryad Digital Repository,” accessed 07/05/2015. [Online]. Available: http://www.datadryad.org/

[2] M. Diepenbroek, H. Grobe, M. Reinke, U. Schindler, R. Schlitzer, R. Sieger, and G. Wefer, “PANGAEA – an information system for environmental sciences,” Computers & Geosciences, vol. 28, 2002, pp. 1201–1210.

[3] “Scientific Data,” accessed 07/05/2015. [Online]. Available: http://www.nature.com/sdata/

[4] “Earth System Science Data, The Data Publishing Journal,” accessed 07/05/2015. [Online]. Available: http://www.earth-system-science-data.net/

[5] “Biodiversity Data Journal,” accessed 07/05/2015. [Online]. Available: http://biodiversitydatajournal.com/

[6] J. Kratz and C. Strasser, “Data Publication Consensus and Controversies,” F1000Research 2014, vol. 3:94, 2014. [Online]. Available: http://f1000research.com/articles/3-94/v3

 ISBN: 978-1-61208-415-2

[7] H. Allen, P. Gearan, and K. Rexer, "6th Annual Data Miner Survey," Rexer Analytics, Tech. Rep., 2011.

[8] J. King and R. Magoulas, 2013 Data Science Salary Survey Tools, Trends, what pays (and what doesn't) for Data Professionals, 1st ed. O'Reilly Media, 2014.

[9] J. Kattge, K. Ogle, G. Bönisch, S. Dìaz, S. Lavorel, J. Madin, K. Nadrowski, S. Nöllert, K. Sartor, and C. Wirth, "A generic structure for plant trait databases," Methods in Ecology and Evolution, vol. 2, no. 2, 2011, pp. 202–213. [Online]. Available: http://dx.doi.org/10.1111/j.2041-210X.2010.00067.x

[10] S. Bowers, "Scientific Workflow, Provenance, and Data Modeling Challenges and Approaches," Journal on Data Semantics, vol. 1, no. 1, 2012, pp. 19–30. [Online]. Available: http://dx.doi.org/10.1007/s13740-012-0004-y

[11] T. Lotz, J. Nieschulze, J. Bendix, M. Dobbermann, and B. König-Ries, "Diverse or uniform? Intercomparison of two major German project databases for interdisciplinary collaborative functional biodiversity research," Ecological Informatics, vol. 8, 2012, pp. 10–19. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S157495411100094X

[12] K. Bach, D. Schäfer, N. Enke, B. Seeger, B. Gemeinholzer, and J. Bendix, "A comparative evaluation of technical solutions for long-term data repositories in integrative biodiversity research," Ecological Informatics, vol. 11, no. 0, 2012, pp. 16 – 24, data platforms in integrative biodiversity research. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574954111000987

[13] R. Walls, J. Deck, R. Guralnick, S. Baskauf, R. Beaman, and et al., "Semantics in support of biodiversity knowledge discovery: An introduction to the biological collections ontology and related ontologies," PLoS ONE, vol. 9, 2014.

[14] J. Chamanara and B. König-Ries, "A conceptual model for data management in the field of ecology," Ecological Informatics, vol. 24, 2014, pp. 261–272.

[15] J. Tennison, G. Kellogg, and I. Herman, "Model for Tabular Data and Metadata on the Web," April 2015, accessed 07/05/2015. [Online]. Available: http://www.w3.org/TR/tabular-data-model/

[16] "Guidelines for the use of Observations & Measurements and Sensor Web Enablement-related standards in INSPIRE Annex II and III data specification development," 2014. [Online]. Available: http://inspire.ec.europa.eu/documents/Data_Specifications/D2.9_O&M_Guidelines_v2.0.pdf

[17] SDMX Statistical Working Group, and SDMX Technical Standards Working Group, "Guidelines on Modelling a Statistical Domain for Data Exchange in SDMX," 2015, accessed 07/05/2015. [Online]. Available: http://sdmx.org/

[18] "Biodiversity and Ecosystem Functioning Data," accessed 07/05/2015. [Online]. Available: https://github.com/befdata/befdata

[19] K. Nadrowski, K. Pietsch, M. Baruffol, S. Both, J. Gutknecht, H. Bruelheide, H. Heklau, A. Kahl, T. Kahl, P. Niklaus, W. Kröber, X. Liu, X. Mi, S. Michalski, G. von Oheimb, O. Purschke, B. Schmid, T. Fang, E. Welk, and C. Wirth, "Tree Species Traits but Not Diversity Mitigate Stem Breakage in a Subtropical Forest following a Rare and Extreme Ice Storm," PLoS ONE, vol. 9, no. 5, 05 2014, p. e96022.

[20] "Exploratories for Large-scale and Long-term Functional Biodiversity Research," accessed 07/05/2015. [Online]. Available: http://www.biodiversity-exploratories.de/startseite/

[21] "The Biological and Chemical Oceanography Data Management Office," accessed 07/05/2015. [Online]. Available: http://www.bco-dmo.org/

[22] "ITIS - Integrated Taxonomic Information System," accessed 07/05/2015. [Online]. Available: http://www.itis.gov

[23] The Knowledge Network for Biocomplexity (KNB), "Ecological Metadata Language (EML)," accessed 07/05/2015. [Online]. Available: https://knb.ecoinformatics.org/#external//emlparser/docs/index.html

[24] "Access to Biological Collection Data (ABCD)," accessed 07/05/2015. [Online]. Available: http://wiki.tdwg.org/ABCD

[25] J. Madin, S. Bowers, M. Schildhauer, S. Krivov, D. Pennington, and F. Villa, "An ontology for describing and synthesizing ecological observation data," Ecological Informatics, vol. 2, 2007, pp. 279–296.

[26] Y. Shu, D. Ratcliffe, M. Compton, G. Squire, and K. Taylor, "A semantic approach to data translation: A case study of environmental observations data," Knowledge-Based Systems, vol. 75, 2015, pp. 104–123.

 ISBN: 978-1-61208-415-2

Context-Aware Healthcare Dataset - A Case Study from Pakistan

Shahid Mahmud
Faculty of Engineering and Computing
Coventry University,
United Kingdom
Email: mahmuds4@uni.coventry.ac.uk

Rahat Iqbal
Faculty of Engineering and Computing
Coventry University,
United Kingdom
Email: r.iqbal@coventry.ac.uk

Faiyaz Doctor
Faculty of Engineering and Computing
Coventry University,
United Kingdom
Email: faiyaz.doctor@coventry.ac.uk

***Abstract*—This paper presents a context-aware healthcare dataset that has been designed to understand and monitor the health-shocks in Pakistan. Based on the socio-economic, cultural, and geographic norms, a user study based on questionnaire comprising of 47 features was carried out. In total, $1,000$ households belonging to 29 villages in rural areas participated in this user study. The purpose of this research is to monitor health-shocks in a community using data visualization and predictive modelling. We envisage that this study will provide insight into the relationships between socio-economic, demographic, and geographical conditions impacting health issues.**

Keywords–Context-aware; socio-economic; cultural; geographical; data visualization; and predictive modelling.

I. INTRODUCTION

Generally, healthcare systems are evaluated based on three main factors: quality, cost, and accessibility to health-care, known as "The Iron Triangle of Health-Care" as shown in Fig. 1. In case of an effective health-care system, there should

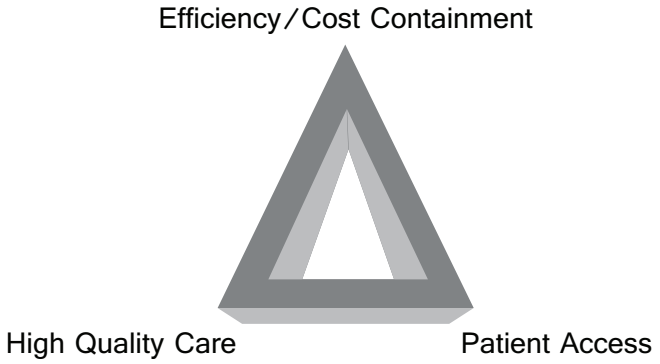


Figure 1. The Iron Triangle of Health-Care [1].

be a balance between all the three components, i.e., the iron triangle should be an equilateral triangle, with each angle of 60° [1]. However, in practice, any effective health-care system can only optimize two of the three factors. For instance, to achieve higher access and quality, its associated cost will increase [2]. Furthermore, these factors highly depend on the socio-economic, geographic, and cultural norms. Especially, in order to understand the health-shocks situation of any third world country, there is a need to understand socio-economic, geographic, and cultural norms of that origin. By health-shocks, we mean critical illness of families, principle bread winner and its socio-economic after-effects on individual, family, society and various governance levels [3].

In this regard, a lot of research work has been done to understand the reasons and effects of health shocks in the developed and developing countries [4]–[9]. In [7], different socio-economic factors and their impacts were studied. It was reported in [7] that 63.8% of health expenditure was out-of-pocket, i.e., from the pocket of patient, which resulted in financial losses. In [10], health related "hardship financing" for poor households in an Indian town Orissa was studied. The authors investigated factors influencing the risk of hardship financing with the use of a logistic regression. It was observed that in rural areas, most of the households were facing financial hardships due to indirect and/or long-term costs of health-care. In Orissa, 80% of spending on health-care was out-of-pocket of the households for which they either borrow money at higher interest or sell their assets.

From the various studies [10]–[12], it is quite evident that the unpredictable timing of health issues and immediate need for large funds for health-care in addition to the distance to health facilities could increase the risk of hardship financing.

In this paper, we have tried to understand the health-care system of Pakistan and how the socio-economic, geographical and cultural norms are affecting the health of almost 200 million Pakistanis, especially those who belong to rural and tribal areas. For instance, women in rural and tribal areas of Pakistan are not allowed by their men to consult a male doctor during pregnancy which results in higher infant mortality rates (IMR) and maternal mortality rates (MMR). In Pakistan, IMR which is a count of the number of infants that die before their first birthday in every thousand infants was 80 at the start of this century. Table I shows the IMR of Pakistan in comparison to other countries. Currently, there are only 25 countries that have a higher IMR than Pakistan.

Another useful measure for assessing children's health is their weight. Experts have figured out a scale that lists out the appropriate weight for healthy children at any given age. In 2001, percentage of underweight children who were less than five years old was 32% in Pakistan. During the millennium development goals, government of Pakistan has vowed to reduce it to 20%, by 2015. In developed countries, such as Japan, this percentage is less than one.

Similarly, MMR which is considered as a basic measure for assessing the health of the mothers in any given region was 490 in Pakistan during 1990. Another important factor that is directly related to IMR and MMR is "appropriate pregnancy

 ISBN: 978-1-61208-415-2

TABLE I. IMR of different countries [15]

Countries	1990	2010
Sweden	6	2
England	8	5
Malaysia	15	5
United States	9	7
Turkey	66	14
Sri Lanka	26	14
Saudi Arabia	36	15
China	38	16
Iran	50	22
India	81	48
Pakistan	96	69

spacing". Generally, it is recommended to maintain a 2.5 to 3 years gap between pregnancies which is vital for the health of both mother and child. However, such spacing is only possible with proper awareness, equipment and its availability, and birth control. According to [14], around 27% of the Pakistani couples who preferred to use some sort of birth control did not find it available in their local region.

II. Anatomy of Dataset

To understand the health shocks and its causes, we collected a dataset of $1,000$ households from the district Haripur with the help of Begum Mahmuda Welfare Trust hospital (BMWT). Haripur district is in the Hazara region of Khyber Pakhtunkhwa province of Pakistan. It is located in a hilly plain area at an altitude of around 610 meters above sea level with an estimated population of a million in 2009 [13]. In district of Haripur, there are about 39 hospitals and basic health centers, and 10 dispensaries. Furthermore, there is only one bed for every $1,516$ people [13]. In contrast to Haripur, there is one bed for every 100 people in the developed countries of the world. Moreover, in district of Haripur, IMR is 66 whereas overall MMR in the province of Khyber Pakhtunkhwa is 275.

Based on "patients to bed ratio", Pakistan is ranked at 178 out of 194, internationally. Furthermore, in comparison to developed countries where there is at least one doctor for every 712 people, Pakistan has only one certified doctor for every $1,230$ people.

A. Survey Features

In order to find the ground realities of health-care facilities in rural areas, we have collected the information about age, marital status, sex of the household head, their involvement in the labour force, education of children, financial and water resources, access to health facilities, schools and clean water, effects of climatic changes, effects of shortage of basic facilities like fuel, food, money for treatment of illness or fertilizer for crops, and waste disposal trends.

B. Ethical Considerations

Based on the cultural norms of this region, five ethical concerns were short-listed that we wanted to confirm while conducting the study. These six ethical concerns includes: voluntary participation, no harm to respondents, anonymity and confidentiality, identifying purpose, sponsor, analysis and reporting. Throughout the survey, these guidelines were followed strictly.

C. Survey Questionnaire

A printed questionnaire was used to obtain the survey information from the responders directly. Here, we preferred questionnaire over the interviews as questionnaires are more systematic, less prone to personal biases and transcription errors. Moreover, it offers more comfort to the responders as they can fill it in their own private settings with discretion. In our study, questionnaires were in national language of Pakistan, i.e., Urdu, in order to enable the villagers who does not understand English to be able to answer the questions. Furthermore, assistance was also provided to the participants who could not read or write. The questionnaire was divided into two sections, i.e., section A and section B.

Section A aimed at gaining demographic data such as age, level of education, income and gender whereas Section B is more concentrated on the living standard of the participants and effects of health shocks on their families. Furthermore, as the targeted population was too large to survey, so accidental sampling was used in identifying the participants. Here, "self-report" was used a responding mechanism where people voluntarily choose to respond to series of questions posed by investigators and are allowed to skip as many questions as they like.

D. Quality Assurance Measures

Different quality measures were also adopted from responder's perspective, i.e., understandability, comprehensiveness, and acceptability of the survey forms. We have used quite a few methods for quality assurance, including *sample testing* by asking few expert interviewers and responders to fill out the survey form and incorporating their suggestions, cognitive testing, by interviewing the responders to visualize the questions and reiterate them in their own words, behavioural testing by altering the questions and measuring the difference in responder's answers to find out how the wording in questionnaire will affect the overall answer of the responders, special probing, by explaining the intent of the questions to the responders in local language, so the responders would not take an infinite amount of time to answer the questions, experts opinion, by sharing the questionnaire with experts for their valuable feedback and suggestions, compare and contrast, by measuring the questionnaire against pre-existing surveys and their responses in order to see what value additions can be made, what best practices can be used, and what errors can be avoided.

E. Pilot Study

In the pilot study, a multidimensional survey questionnaire was distributed among 300 families living in the proximity of Haripur district. During the survey, geo-coordinates of 29 villages of district Haripur which participated in the survey were also noted as shown in Fig. 2. One of the primary objectives of the pilot study was to refine the survey questionnaire based on its feedback. During the pilot study, some families came to the hospital and/or local clinics to fill the questionnaire while others were contacted at their homes. The questionnaire was given to the household heads who were older than 21 years with one or more family members living with them, who were mentally stable, and were willing to participate in the survey. Furthermore, there was no race, religion, and gender discrimination during the survey activity.

 ISBN: 978-1-61208-415-2

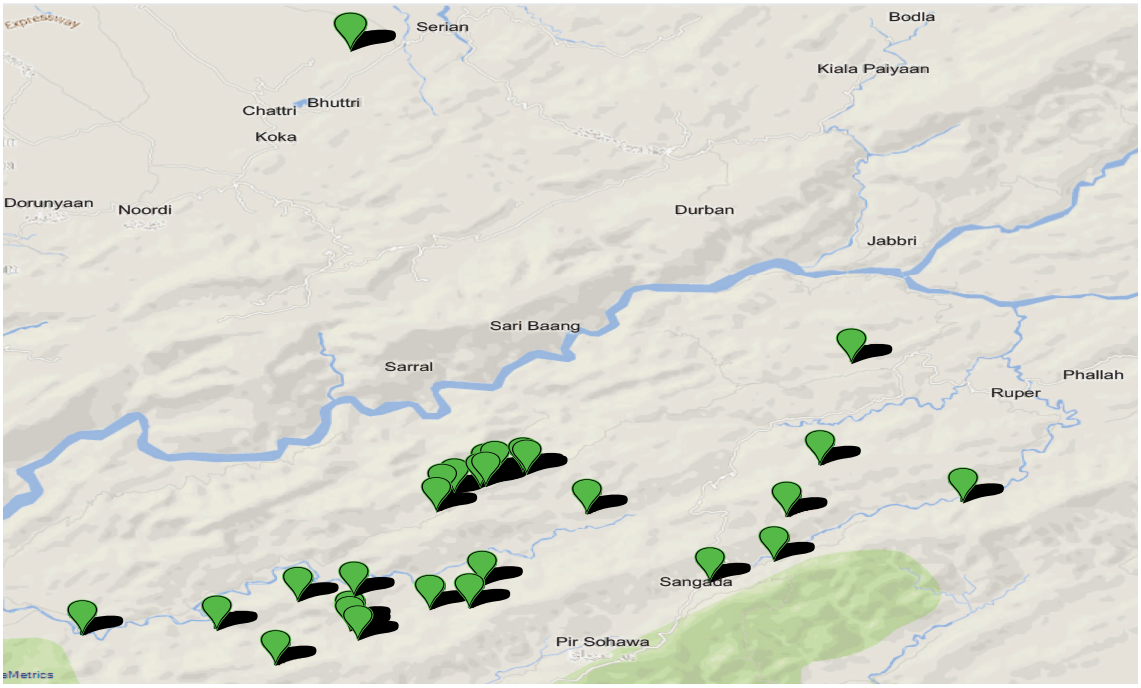


Figure 2. Geo-coordinates of the villages who took part in the survey.

During this pilot study, we also convinced the village elders to actively participate in the survey and to provide us a good insight about available health facilities in their villages. With the active participation of the village elders, we met number of families and explained them the purpose of this survey and the changes that it can bring in their daily lives. As a result, 300 households participated in the pilot study. We also made sure to maintain the confidentiality and integrity of the households by masking their names in the dataset.

As a result of the pilot study, we have found numerous information that have helped us to refine the survey, for example, most of the subjects have a hard time getting clean water or do not have a proper toilet facility. Almost all participants are single with at least two adults living with them. Students from participating families have to travel for at least 20 minutes to get to school. Minor illnesses occurred at least twice a year while major diseases occurred 3-5 times a year in most of the households. Most families fall short of money if anyone in the family falls ill or has an injury. Stone and mortar is mostly used as the basic construction material of external walls while ceilings are mostly made of thick wood. People mostly don't have any toilet facility at home. Waste food, water and garbage are mostly disposed off near homes. Almost every time during the year people have to rely on an irrigation canal for water source. Participants rely mostly on land and use it for agriculture or livestock. Most common hardships faced are loss of job or losing a house.

Furthermore, based on the feedback, we added some other information in the survey that affects the overall health-care system in Pakistan, namely: distance to the health units, number of basic health units, costs associated with travelling, accessible routes to the health facilities, vaccination, transportation, sewerage system, awareness, and water resources, just to name a few. Fig. 3 shows the modified iron triangle that fits well to the socio-economic, geographic and cultural norms of our region.

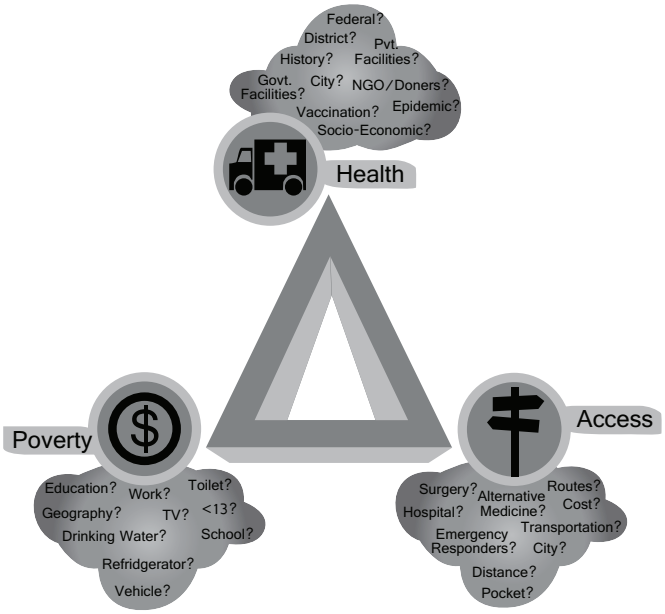


Figure 3. Factors affecting the Health-Care system of Pakistan.

F. Data Collection

At the end of the pilot study, all the changes were incorporated in the questionnaire. During the survey, we followed a structured approach in which all the responders were given the same possible choices and all questions were presented to the responders in the same order, i.e., instructions and

 ISBN: 978-1-61208-415-2

explanations were fixed. The questionnaire was filled by $1,000$ households. The head of the household was given a brief account of the research and its importance and the support of the administrator. Finally, we provided a telephone number to the village elder for anyone with questions or who may need assistance in completing the questionnaire at home. All participants were provided a comfortable environment and privacy. The questions were in easy to understand language. In case of unanswered questions, questionnaire was brought back to the participants to know the reason for not answering the questions. The reason was then noted on the questionnaire next to that question. In total, BMWT dataset comprises of $1,000$ households and for each household there are 47 features that plays a vital role in the health-care system of Pakistan. Table II shows the feature set of BMWT dataset.

TABLE II. Features of BMWT Dataset

Sr. No.	Features
1	Subject ID
2	Contact Number
3	Gender
4	ID Card No.
5	Marital Status
6	Age
7	Tehsil
8	Union Council
9	Village
10	GPS Coordinates
11	Adults living in house for more than 9 months in a year (Female)
12	Adults living in house for more than 9 months in a year (Male)
13	Adults earning
14	Distance of schools in Kilometers
15	Distance of schools in Minutes
16	Frequency of minor disease/year
17	Frequency of severe disease/year
18	Distance of basic health unit in Kilometers
19	Distance of basic health unit in Minutes
20	Distance of Hospital in Kilometers
21	Distance of Hospital in Minutes
22	Mid Wife During Birth
23	Distance of Vaccination Center
24	Polio Drops
25	Fatalities During Birth
26	Nature of Walls of House
27	Nature of Ceiling of House
28	Resistance of House against Severe Weather
29	Toilet Facility
30	Disposing off of food
31	Disposing of garbage
32	Disposing off of water
33	Dental Hygiene
34	General Hygiene
35	Water Source (most of the year)
36	Water Source (in dry weather)
37	Time Duration for collecting water for one day
38	Agricultural Land (in canals)
39	Expenses of Manure for Land
40	Domestic animals - Buffaloes/Cows
41	Domestic animals - Goats
42	Ownership of Land
43	Expected Problems (1st, 2nd and 3rd preference wise)
44	Solutions to expected problems
45	Duration for reconstruction of House in case of destruction (in months)
46	Shortage of food
47	Debts

III. DATA VISUALIZATION

Data visualization has been done using the purpose built "HexChange" tool that was developed in C# (We are working on its web-version which will be made publicly available). In BMWT dataset, 47 features of $1,000$ households belonging to 29 villages of Haripur district were captured as shown in Fig. 4. Here, for the open-ended questions, we opted for the quantitative content analysis which is a formal, systematic, and objective process used in describing and testing the relationship and their causal interactive effects among variables.

Here, it is worth mentioning that all the features mentioned in Table II are highly dependent on each other. For instance, there is a positive relationship between distance to basic health units (BHUs), percentage of debts, toilet facilities, and frequency of major illnesses. Due to debts and distance to BHUs, minor illness turns into major.

Table III and Table IV shows the distance to BHUs and hospitals, respectively. It is clear from Table III that in Barkot, on average, each patient needs to cover a distance of 12.81 kilometres in order to reach a BHU with a standard deviation (std.) of 9.91 kilometers. Similar results can be seen in Table IV, where each patient on average travels 24.01 and 25.62 kilometers from Barkot and Muslimabad to reach hospitals.

TABLE III. Distance to BHUs in KMs

Union Councils	Min.	Max.	Mean	Std.
Barkot	0	34	12.81	9.91
Jabri	2	20	8.6	5.55
Musalimabad	9	18	15.27	2.63
Muslimabad	0	34	10.94	7.23
Najafpur	3	30	8.26	4.07

TABLE IV. Distance to Hospitals in KMs

Union Councils	Min.	Max.	Mean	Std.
Barkot	0	51	24.01	7.51
Jabri	10	34	27.21	5.56
Musalimabad	14	34	20.53	5.15
Muslimabad	0	36	25.62	7.92
Najafpur	10	120	18.23	11.18

In district of Haripur, mean poverty score is 29.94 [16]. This poverty score is reflected in Table 5, where 49.55% of responders from Barkot are under the debt of more than 500,000 PKR.

TABLE V. Majority of people are in debt with moderate to large sums of money

Union Councils	Don't Know	No Debt	1K to 200K	200K to 500K	>500K
Barkot	5.04	4.75	15.13	25.52	49.55
Jabri	6.35	1.59	52.38	39.68	0.00
Musalimabad	40	6.67	46.67	6.67	0.00
Muslimabad	1.05	12.87	11.60	29.75	44.73
Najafpur	4.72	8.49	21.70	65.09	0.00

Due to higher distance of BHUs and hospitals in Barkot, people have to pay much more in order to reach the health facility. Here, cost of travelling is more or equal to the cost of treatment which results in higher debts and severe illness. Table 6 shows the frequency of major/severe diseases. Here, major/severe disease is defined as an injury which requires two or more days of bed rest or hospital admission. It also includes disability.

Table IV-VI highlight the relationship between distance to BHUs and hospitals, debts, and major illness which supports our observation that socio-economic, geographical and cultural

 ISBN: 978-1-61208-415-2

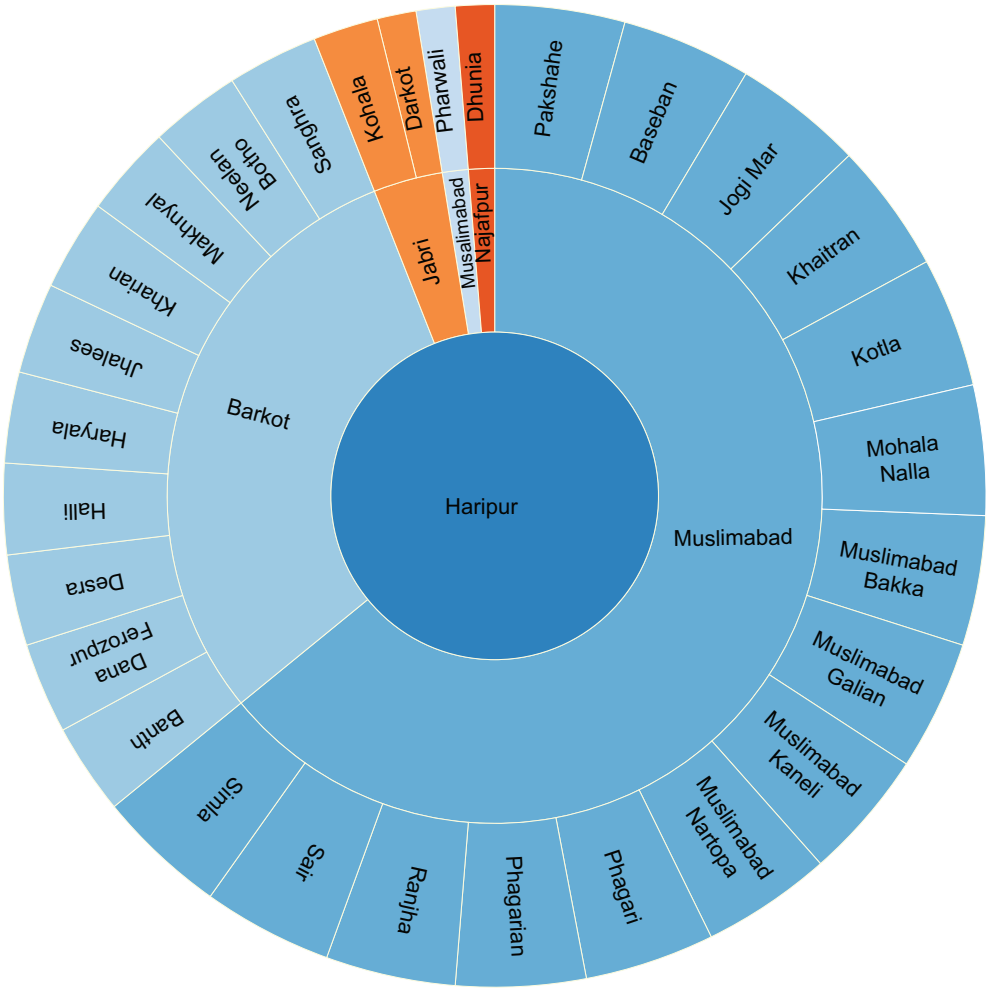


Figure 4. Villages that participated in the survey. Inner most circle represents the district Haripur whereas the outer circle represents the union councils and the outer most circle represents their corresponding villages.

TABLE VI. Frequency of Major Disease in Percentage

Union Councils	Never	Yearly	Bi-Monthly	Monthly	Bi-Weekly	Weekly	Daily
Barkot	5.06	13.99	7.44	20.83	14.58	24.7	13.39
Jabri	1.59	17.46	4.76	11.11	1.59	50.79	12.7
Musalimabad	13.33	20	6.67	13.33	6.67	20	20
Muslimabad	9.92	20.46	11.39	20.89	10.13	12.66	14.56
Najafpur	12.26	10.38	6.6	19.81	14.15	25.47	11.32

norms highly affects the health-shocks, especially in the rural and tribal areas.

Furthermore, it is interesting to see the houses with toilet facilities have higher rate of minor and major diseases in comparison to houses with no toilet facility as shown in Fig. 5. One of the main reasons was access to water resources and poor sewerage system. Here, it is worth mentioning that the time required for household to collect water for one day usage is almost 4 hours as in some cases, it takes a women 1.5 to 2 hours to reach the source. Same amount of time is required to carry that water back home. Especially, in case of a family with two to three children, it requires more than 5 or 6 buckets of water at least for a day.

IV. Conclusions

Currently, there is no publicly available dataset that can help to understand and monitor the health-shocks in Pakistan. Such kind of surveys and datasets can be helpful to government - who would use this dataset and resulting analysis to form policies, to general practitioners and NGOs, in order to start community based health programs. This dataset is our first initiative to analyse and understand the health-care system and health-shocks, especially in rural and tribal areas of Pakistan (For those who are interested in BMWT dataset, please contact: mahmuds4@uni.coventry.ac.uk). Our proposed future work is to apply machine learning techniques to an extended dataset sampled from a larger population to develop a predictive model of health-shocks that forms part of a framework which

 ISBN: 978-1-61208-415-2

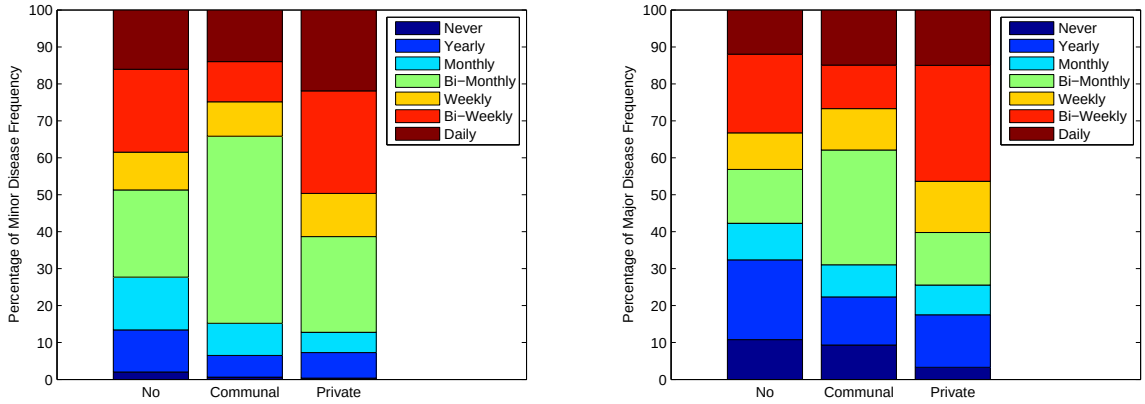


Figure 5. (a) Frequency of minor disease versus toilet facility. Here, communal represents a toilet shared by more than 3 people. b) Frequency of major disease versus toilet facility.

uniquely accounts for the cultural and traditional norms of this part of the world. The aim of this data intelligence driven framework will be to provide tailored and informed health-care analysis to stakeholders towards facilitating a national agenda of health-care reform.

REFERENCES

[1] K. William, "Medicine's Dilemmas," in *New Haven, CT: Yale University Press, 1994.*

[2] L. R. Burns, "A System Perspective on India's Health-Care Industry," in *global economy, health-care reform, India, World Development, July, 2014.*

[3] S. Mahmud, R. Iqbal, and F. Doctor, "An Integrated Framework For The Prediction Of Health Shocks," in *Proceedings of The 2nd International Conference on Applied Information and Communications Technology (ICAICT), April, 2014.*

[4] R. M. Townsend, "Risk and insurance in village India," in *Econometrica, vol. 62 (3), pp. 539591, 1994.*

[5] A. Kochar, "Explaining household vulnerability to idiosyncratic income shocks," in *American Economic Review, vol. 85 (2), pp. 159-164, 1995.*

[6] E. Skoufias and A. Quisumbing, "Consumption Insurance and Vulnerability to Poverty: A Synthesis of the Evidence from Bangladesh, Ethiopia, Mali, Mexico and Russia," in *The European Journal of Development Research, vol. 17 (2), pp. 24-58, 2005.*

[7] M. R. Howlader, "Analysing the socio-demographic variables impact on health status of Bangladesh," in *social science research network, 2013.*

[8] R. Iqbal, N. Shah, A. James, and J. Duursma, "ARREST: From work practices to redesign for usability," in *Journal of Expert Systems with Applications, vol. 38 (2), pp. 1182-1192, 2011.*

[9] F. Doctor, R. Iqbal, and R. Naguib, "Fuzzy Ambient Intelligent Agents Approach for Monitoring Disease Progression of Dementia Patients," in *Journal of Ambient Intelligence and Humanized Computing (AIHC), vol. 5 (1), pp. 147-158, 2014.*

[10] E. Binnendijk, R. Koren, and D. M. Dror, "Hardship financing of health-care among rural poor in Orissa, India," in *BMC Health Services Research, 2012.*

[11] Robert Wood Johnson foundation, "Changes in health-care financing and organization," in *2009.*

[12] R. Victor, F. Jeffrey, and E. Harris, "Review: Who Shall Live? Health, Economics, and Social choice," in *The Bell Journal of Economics, vol. 7 (1), pp. 340-343, 1976.*

[13] Government of Pakistan, "Khyber Pakhtunkhwa Health Sector Situation Analysis," in *December, 2010.*

[14] W. Hameed, S. K. Azmat, M. Bilgrami, and M. Ishaqe, "Determining the factors associated with Unmet need for family planning: a cross-sectional survey in 49 districts of Pakistan," in *PJPH, vol. 1 (1), 2011.*

[15] –, "UN Inter-agency Group for Child Mortality Estimation (UNICEF, WHO, World Bank, UN DESA Population Division)," *2014.*

[16] Durr-e-Nayyab, "Population Dynamics in Pakistan: An Analysis of the BISP Poverty Score Survey," *2010.*

 ISBN: 978-1-61208-415-2

Query Acceleration in Multimedia Database Systems

Ramzi A. Haraty and Rawa Karaki
Department of Computer Science and Mathematics
Lebanese American University
Beirut, Lebanon
e-mail: rharaty@lau.edu.lb

***Abstract*--With the increasing popularity of the World Wide Web comes the enormous increase in stored digital contents, which could challenge users to search and use the multimedia data efficiently. This work focuses on hastening techniques for efficient retrieval of multimedia data. In this paper, we exploit the use of bit-vectors to accelerate queries in multimedia databases. We use a compressed bit-vector to minimize the amount of data cashed on disk; thus, reducing the amount of memory and time needed to execute queries. We also compare our scheme with other related strategies.**

***Keywords-multimedia; bit vectors; query accelerations*.**

I. INTRODUCTION

Multimedia databases have become one of the puffs in computer science technology. It is a recent evolution of the Internet and data warehousing. Many authors wrote about the evolution of multimedia databases and ways to implement it [1][2]. Multimedia is a mix of multiple mediums - images, sounds, music, audios and videos etc. As long as the development of the Internet and computer technology continues, multimedia files will appear more and more in many applications. For that reason, it is important and significant that the data files of multimedia objects be arranged, ordered and categorized so we can simply access them at any time. Therefore, multimedia databases are the necessary tool to handle and support these enormous multi-media object files.

A multimedia database is a type of database that is similar to all other database types except that it contains multimedia files in its collection. To organize and manage multimedia data files, a multimedia database management system is needed. It is a program that runs and directs the collection of media files and allows entry for end users to retrieve multimedia files or objects. In general, multimedia databases hold images, audio, video, animations and many other file forms. All files or data are saved as binary forms in the multimedia database.

Multimedia database implementation differs from regular database implementation in the design of the media objects and files where the files are kept and stored. Different characteristics of multimedia data represent the diversity of the data since they are complex--composed of audio-visual data. Research shows that objects in multimedia data are complex and involve a chained structure that can hold a connection between them [3][4]. Static media, such as text, graphics, and images, are time-independent like. For instance, image files do not have time-related action because there is no connected time factor. Video files, on the other hand, are dynamic, and have both time and dimensional dependency. This is due to the fact that a video is composed of multiple ordered image frames which associate to form the video file.

In this paper, we use a compressed bit vector for multimedia data retrieval to select files from a database more efficiently. The method facilitates rapid searching of multimedia data objects in a multimedia database. A single bit vector is used to determine matches for the main query, returning a reduced set of multimedia objects instead of the entire multimedia data object; thereby greatly reducing the query search time, increasing the efficiency of the process by allowing the bit-level operations and minimizing the cost and amount of data transferred. The execution time is exactly proportional to the size of input. The algorithm complexity is of order O(n).

The rest of this paper is organized as follows: Section 2 provides a brief explanation of multimedia database management systems. Section 3 presents related work. Section 4 presents the compressed bit vector algorithm and its execution results. Section 5 gives the conclusion and future work.

II. MULTIMEDIA DATABASE MANAGEMENT SYSTEMS

With the evolution of Internet and computer users, multimedia data text, graphics, and images a greater effect on our daily lives. That is why finding a new technique to easily retrieve enormous multimedia information and a file, at any point of time, is in high demand. Any multimedia object can be generally described as a group of extended, shapeless

 ISBN: 978-1-61208-415-2

series of bytes. These objects are called BLOBs (Binary Large Objects). BLOB files are usually very large in size; for this reason, database management systems provide particular maintenance to insert, delete, modify or retrieve BLOB objects from database.

Modern databases are frequently capable of storing BLOBs and CLOBs (Character Large Objects), as columns in their tables. Data stored in a BLOB column can be accessed using connectors and manipulated using client-side code. Reading a BLOB from the database is a slow task considering the size of a multimedia object. A BLOB can contain as much as four gigabytes of data for each field. Multimedia database systems are thus required to provide an efficient cache of the BLOB files, but this is not sufficient for multimedia implementation maintenance. Therefore, a query of a prolonged continual series of bytes is restricted to a matching pattern and reorganization of a BLOB multimedia object may return zero results due to missing constructional information. Even if it can be realized, to draw out information of the object in realistic time, for example working with pattern identification techniques, would be unrealistic. Thus, a multimedia database system should keep an analytical structure of the BLOB files. Multimedia objects can be saved in smaller parts to allow easier retrieval of BLOB objects based on content. Multimedia data is sizeable and have an impact on the retrieval, insertion and manipulation of multimedia data files. The large amounts of data to be processed can be checked against those that need to be processed. Table I illustrates the enormous sizes of data for media files of different types.

III. RELATED WORK

Querying and retrieving information in multimedia databases differs from traditional databases [5][6]. A fairly straightforward search can be done in alphanumeric databases. Multimedia databases contain pictures and different complex multimedia data objects; thus, the database is not easily indexed, classified and retrieved [7]. How is it possible to retrieve a picture with a cup of water or a horoscope sign? Those shapes are difficult to recognize. Some retrieval classes for multimedia databases include:

- Retrieval by Browsing (RBR): Browsing multimedia objects to retrieve the best matching file. For example, using a simple interface to let users browse small images known as “thumbnails” to pick the image that matches the query.
- Retrieval by Metadata Attributes (RMA): Designing a query that addresses the Meta and logical characteristics. For this purpose, any media file is stored with information describing the file. For example, we will not query an image with a bird but we will address our search to find which media handles the keyword ‘bird’ as its meta information.
- Retrieval by Shape Similarity (RSS): It is a type of retrieval based on media content. Searching in a multimedia database based on shape similarity of the file. For example, retrieve all the images that contain a circle.
- Retrieval by Content Attributes (RCA): Query is sent with enough detail describing the file to be retrieved. For example, retrieval of all images that contain a specific celebrity.

In this paper, we focus on the RBR and RMA since they are the most widely used retrieval classes in multimedia databases.

A. The Retrieval by Browsing

A user who requests the search for a specific file uses terms and details to illustrate the retrieval system. Then, the software matches the query with existing matching objects and returns a list of files to the end user for examination. The end user then considers the retrieved files and picks items that exactly match his needs. This type of retrieval works best in finding the exact requested file, but multiple problems appear with its implementation:

1. End users find it hard to formulate queries.
2. Queries may return only unwanted files and result in too many suggested unwanted matches.
3. Query terms are not properly valued.
4. Multiple forms of image and audio files that need conversion.

TABLE I. SAMPLE MEDIA TYPES, FORMATS, AND RELATED DATA VOLUMES AND TRASFER RATES [5].

Media Type	**Sample Format**	**Data Volume**	**Transfer Rate**
Text	ASCII	1MB/ 500 pages	2KB/page
B/W Image	G3/4-Fax	32MB/500 images	64KB/page
Color Image	GIF, TIFF, JPEG	1.6GB/500 images 0.2GB/500 images	3.2MB/image 0.4MB/image
CD-music	CD-DA	52.8MB/5 minutes	176KB/sec.
Consumer Video	PAL	6.6GB/5 minutes	22MB/sec.
High quality video	HDTV	33GB/5 minutes	110MB/sec.
Speech	m-law, linear; ADPCM, PEG audio	2.4 MB/5 minutes 0.6MB, 0.2MB/5 min.	8KB/sec.

 ISBN: 978-1-61208-415-2

Different authors have proposed that browsing, which uses the human recognition capabilities, can control and solve the above difficulties [8][9]. Though, the retrieval by browsing is suggested to be a direction solving many problems in multimedia retrieval and handling multimedia systems, but it is logically seen as a difficult and time inefficient task for humans to solve [10].

B. The Retrieval by Metadata Attributes

Generally, human beings have the power to retrieve and correlate information efficiently. It is unfeasible to search millions of data by simply "staring" in order to assemble diverse documents, which may involve texts, videos, audio and images files, either alone or as multimedia items. Thus, we seek a simple technological multimedia search based on known information of the file.

Metadata are data about data. Metadata can describe any data using different categories: quantity, quality, materials, shape and different properties of the data as tools to find, understand and access the data files. Metadata details can aid users to have an explanation about the data being searched in multimedia databases. The picture itself describes nothing more than an ordinary image with colours. Without having the metadata description associated with the picture, it will be out of question for machines to know the properties of this picture. For example, if we would like to know when and where this picture was taken, or its resolution etc., we turn to Metadata. All this information does, is provide a key that aids in specifying the properties of the image to be used in many applications [11].

The Metadata model requires descriptive information of the content, combined with contextual information, saved in the multimedia database in reference to the multimedia object, and used as an information tool for browsing with a point of association of a specific media. Descriptive information is valuable for searching a multimedia object, and is of major importance when contacting explored results where the attribute, such as a photographer name, a singer name or date, are applied to choose and retrieve the file. The metadata representation of the file is flexible and adopts a multilevel approach for describing the file to permit multiple particles to describe the facts and figures of the file. The metadata model may be unusable to work on a single level in describing a media file with multiple classes of representation [12]. For an image, multiple descriptive data are associated with saved image snaps that can provide accommodation in the model. For a video file in a broadcasting station, information could be automatically produced for each shot or segment that describes the scene.

Lord and Pratt reported a technique of retrieving data from a BLOB data warehouse using SAS as the data analysis tool [13]. The data warehouse architecture requires storing summary data in traditional database relational databases and storing raw chip data in a multimedia database BLOB data type. With this BLOB data type, many opportunities have opened up for experimenting with various methods of retrieving data. Since the databases are fragmented among multiple machines (due to the large data volumes), and to make it easy to register a structure that is required to access the inner parts of the BLOBs, a machine is set aside specifically to direct the client applications and SQL users to the machine where the required data resides. This machine also provides the information necessary to extract parts of the BLOBs. We refer to this machine as the application director. At the database end, the objects would be too large to be practical. With data volumes in the hundreds of gigabytes, adding descriptive information into the records would explode the data storage requirements beyond reasonable limits. Objects also allow us to store large numbers of data values.

After the storing of the object, we have to specify how to access this object. This is where the registry comes in. The registry is a set of tables that define the type of object; in this case the type is defined by the application, (not necessarily a database data type) and the contents of the object. Each object is comprised of elements that have a name, type, and length. All of this information is stored in the registry. The query looks into the objects and extracts that element, returning it as a column in the user view. An example of a query is as follows:

```
SELECT LOT, WAFER, CHIP, SETELEMENT
    (OBJECT1, D_VAL1)
FROM DB.TABLE1
WHERE LOT = '123456789' AND WAFER = 'ABCDEF'
```

This query gets the BLOB object1 in the database from the TABLE1 table and finds the D_VAL1 element in each object, returning it as a column in the table.

Srivastava and Velegrakis [14] described that several metadata management tools consider the metadata as an integral part of the data, which means that metadata cannot be retrieved without also retrieving the data with which it is associated. The authors showed that storing the metadata in independent tables, associated to the data through the q-values, allows them to be queried and retrieved independently. For instance, if a user would like to know the sources that have been used to collect info of a file, he can simply query the metadata table alone.

IV. THE COMPRESSED BIT-VECTOR FOR MULTMEDIA DATA RETRIEVAL

The existence of an enormous volume of media data files questions the aspects of the management of multimedia objects and the problem of implementation. Typically, queries in multimedia database are multidimensional and have complex selections. Users that request specific queries in multimedia databases usually find it hard to find answers to all requirements. Due to these characteristics, bit-vector indexing techniques have shown promising results for processing multimedia databases [15]. A significant advantage of the bit-vector technique is that complex logical selection can be performed very quickly via bit-wise AND, OR and NOT operators. In this paper, we further explore the issues of query acceleration using bit-vectors, and we concentrate on optimizing one of the query operations "Selection,"

 ISBN: 978-1-61208-415-2

which is further discussed with simple queries, and later with more complex queries using the four different types of joins: hash join, inner join, merge join and nested loop join. The space for the compressed bit-vectors works best compared to other techniques.

A bit-vector is a vector or array of data that stocks bits briefly. A bit vector is time composed from the bit values of the collection {0, 1}. Bit-vector is a term applied here to denote a large classification and indexing plan that stocks index as bit sequence. A bit-vector is a bit string in which each bit is mapped to a record ID. A bit in a bit-vector is set to 1 if the corresponding ID has a property "P" and is reset to 0, otherwise. The property "P" is true for a record if it has the value "x" as attribute "X." The query selection can also involve many attributes. Bit-vectors permit vectors of bits to be stocked and handled in the memory set for extended time phases. Bit-vectors can potentially explore bit-level similarity, utilize the data cache to the max, and minimize access to memory. Bit-vectors usually work best in different data forms on reasonable data sets, and on those that are efficient asymptotically [16]. To further improve their effectiveness, we study their compression scheme, which will potentially minimize the area used without expanding the managing time of the query.

Generally, a bit-vector is stocked as a group of bits and the majority of operations on regular bit-vectors are logical bitwise operations. Considering our concerns in using the bit index on huge databases, the main aid is to reduce the sizes of the index. In addition, we aim to efficiently execute logical operations on the compressed bit-vectors. A problem with using uncompressed bit-vectors is their large size and possibly of high expression assessment costs when the indexed attribute has a high cardinality [17]. A single technique to deal with using bit-vectors on high-cardinality attributes problem is to store them in a compressed bit-vector form. Using compressed bit-vectors has multiple advantages that potentially adjust performance: minimized disk space needed to stock the indices, faster reading of the indices from the disk into the memory, and more cached indices in the memory with this compressed form. Several Boolean operation evaluation algorithms, which operate on compressed bitmaps without having to decompress them, might be faster than same operations on the regular bit-vectors. The scheme for compressing data, in addition to transforming data, guides the reducing of enormous volume required. The technique here is to alter the issued multimedia data bit-vector to another modified area to eliminate the redundancies in the real data.

A bit vector "B" of "u" bits can be represented as B[0::u). It can be stored in uH1(B) bits so that the operations can be answered in constant time. We will only save the 1-bits in if the response to the query is true. With this representation of "B," we can access any block of size "b" in constant time, which is sufficient for implementing rank and selecting. In addition, access queries can be answered in constant time, as well.

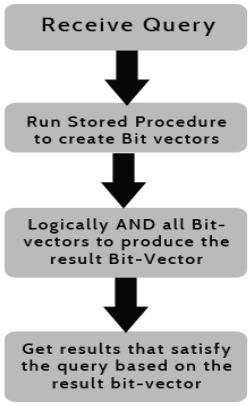


Figure 1. Algorithm workflow.

Decompression is made from the backwards process to retransform and decode the data to its native origin form. This operation generally encounters some data loss, which is a major problem of multimedia applications. Our algorithm ensures negligible loss of data when retrieving information.

Our algorithm compresses bit-vector for multimedia data retrieval and uses these bit vectors to return exact answers to any query in multimedia databases, with any retrieval process used. For example, a specific shape may be compared to a number of pictures in a multimedia database to find a picture or many pictures with the same characteristics. The search may result in either one or more matches found, or no matches at all in a set of objects in the multimedia database.

Figure 1. is an example operation on how a query can be handled in searching for a specific attribute in a multimedia database. First, a receive query operation receives a query item. When a user requests a query in multimedia database with some attribute, a bit vector index is created for each attribute. Each bit vector index indicates whether each of the attributes in the selected database does or does not exist in any of the retrieval strategies used. When a query is received, the bit vector indices associated with each of the selected attribute values are then logically ANDed together to form a single result bit vector index. The result bit vector index identifies a reduced set of accepted IDs of the data table containing the multimedia objects. This reduced set of IDs in the multimedia data objects returned by the bit operations may then be quickly searched using a linear scan to determine a match or matches for the query point. To retrieve resulting matches, we simply select the IDs of the query table that contain a "1" bit in the bit-vector. The stored procedure used in building the bit vector of the specified attributes for any query in multimedia database is depicted in figure 2. For simplicity and straightforwardness, we used the "retrieval by meta and logical attributes" strategy in a real university database.

 ISBN: 978-1-61208-415-2

```
DROP PROCEDURE IF EXISTS  mysql_BitVectorTable //
CREATE PROCEDURE mysql_BitVectorTable ( IN attributeValue
VARCHAR(255))
BEGIN
        DECLARE idSelected    VARCHAR(255);
        DECLARE exit_loop BOOLEAN;
        -- Cursor for select statement
        DECLARE query_cursor CURSOR FOR SELECT id FROM
students where city = attributeValue;
        DECLARE CONTINUE HANDLER FOR NOT FOUND SET
exit_loop = TRUE;
        DROP TABLE IF EXISTS bitvector;
        -- create a new table in database with id and Boolean
   CREATE TABLE bitvector (id VARCHAR(7),bitValue BOOLEAN);
        OPEN query_cursor;
        query_loop: LOOP
                FETCH  query_cursor INTO   idSelected;
                -- save in bitvector
                INSERT INTO bitvector (id,bitValue) VALUES
(idSelected,1);
                IF exit_loop THEN
                        CLOSE query_cursor;
                        LEAVE query_loop;
                END IF;
        END LOOP query_loop;
END //
```

Figure 2. The Create Bit Vector stored procedure.

After the construction of the bit vector, it will be stored in the database as a regular table. Each bit vector contains two fields: The first corresponds to the original table index, and the second contains the bit 0 or 1 referring to the absence or presence of the main query attribute. Each bit vector should contain the same number of indexes as the original table. But to compress our bit vector, we will only save the 1 bits associated with the presence of the query attribute and remove the 0 bits from the bit vector. Thus, the bit vector will contain a smaller number of bits and minimize the response time of the process.

The first experimental query is to select all information and profile picture of students that belong to a specific campus in a specific major. We ran our algorithm on a database table containing multimedia files. We used a traditional database application that uses fixed sized data, but the multimedia size of data can vary dynamically. All unformatted data (mainly text and images) has been handled in this database system through BLOBs. They usually support only a few generic operations, such as reading or writing parts of BLOB. The first table used is the student application table with student images in each record. The table includes more than 510,000 records of student information. The tested query involves retrieving the student images that match certain required parameters. The outcome result will determine the time it took to handle this simple query.

In this simple query, the program indicates that it requires an execution time of 107.334 seconds. This means that there is a need for a method to run queries and return results in a more efficient time. The stored procedure, described above, is used to build the bit vector for the same simple query. A stored procedure is built for every attribute value in the query. After selecting the first attribute, a bit vector table is created and saved in the database. A second bit vector is created for the second attribute. Creating both bit vector took:

$$0.799+1.446 = 2.245 \text{ seconds}$$

Next, we will "AND" all bit vectors created to maintain the final bit vector. Using a time calculator, the retrieval of student images took 3.84 seconds to display on the website. We have also tested our algorithm on different kinds of queries. Other than the simple query noted above, we used two attributes for tables with an index.

We ran our algorithm on simple queries using two attributes for tables without index, then for complex query using hash join, inner join, and nested loop join. To test our algorithm on another more complex query, we will use the "inner join" type. For example, we ran our algorithm with the following query:

```
SELECT id FROM applications
INNER JOIN majors
        ON applications.mjrid = majors.mjrid
WHERE attribute1 = 'a' and attribute2 = 'b'
```

The time it took to build the results of this query in the regular case is: 112.182 seconds. Furthermore, the processing time to display the result is: 3.6691 seconds. The required total time for our algorithm is: 10.73 seconds. The previous results show the efficiency and rapidity of searching of multimedia data using the bit vector algorithm with the metadata retrieval system. Table II shows the time of different kinds of queries with and without applying our algorithm.

To further enhance our algorithm, we wrote it without a stored procedure function. Code that generates the bit vectors stored on the web server functioned as the bit vector. The query selected each attribute alone to retrieve the IDs that match the query results. Then the bit vector was saved in the memory using a key and a value. The key corresponds to the media file ID in the database, and the value corresponds to {0, 1} of the bit vector. To compress our bit vector, we only saved the 1 bits in memory.

TABLE II. EXECUTION TIME OF VARIOUS QUERY STRATEGIES.

Query Type	Running Time Without Bit-Vector Algorithm	Running time With Bit-Vector Algorithm Using Stored Procedure
Query with Attributes For Table With Index	107.33 seconds	6.87 seconds
Query with Attributes For Table Without Index	121.54 seconds	11.28 seconds
Query with Inner Join	112.18 seconds	10.73 seconds
Query with Hash Join	106.53 seconds	5.53 seconds
Query with Nested Loop Join	107.87 seconds	6.71 seconds
Query with Merge Join	107.12 seconds	6.54 seconds

 ISBN: 978-1-61208-415-2

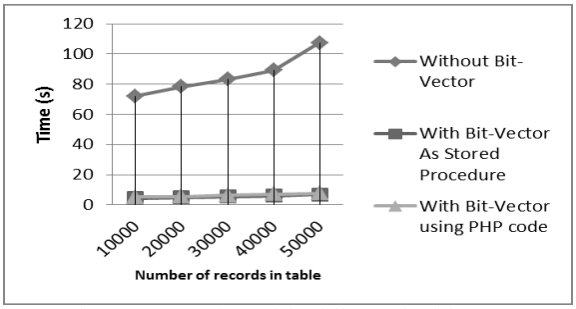


Figure 3. Comparison of the different algorithm.

After saving the bit vectors for each attribute, we added the "AND" or "OR" in the bit vectors according to the query requirements to get the final IDs that respond to the query result. The results are depicted in figure 3.

To calculate the complexity of our algorithm, we defined time taken by the algorithm without depending on the implementation details as our algorithm runs in linear time. The execution time is exactly proportional to the size of input. The algorithm complexity is of order $O(n)$.

V. CONCLUSION

A new strategy is proposed for retrieving multimedia data objects stored in a database. We searched for specific queries selecting objects from a multimedia database such as searching for particular images stored in the database. As a result of the search, either one or more true results are found, or no result exists in the set of objects in the database. Our bit vector for retrieving media files algorithm was proposed and tested on real data. In fact, bit vector indexing techniques have shown promising results for processing multimedia databases. We have explored the issues of query acceleration using bit vectors, and we have concentrated on optimizing "Selection" using the four different types of joins: hash join, inner join, merge join and nested loop join. To optimize the results returned, our method uses a compressed bit vector to save the accepted rows of information. This method guarantees fast and efficient query results. This technique also minimizes the cost and amount of data transferred. Our test results show that the simplest approach towards solving queries in multimedia database is the linear scan. This approach outperformed more complicated approaches.

As for future work, we are currently working on using this compressed bit vector to construct abstractions to be used for more powerful concurrent query analyses in multimedia databases, such as saving repeated queries in existing libraries. This may lead to more efficient and faster query response time.

ACKNOWLEDGMENT

This work was sponsored by the Lebanese American University - Beirut, Lebanon.

REFERENCES

[1] G. Chechik, . Le, M. Rehn, S. Bengio, and D. Lyon, "Large-scale content-based audio retrieval from text queries". Proc. of the ACM MIR'08, Vancouver, Canada, October 2008.

[2] D. Grangier and A. Vinciarelli, "Effect of segmentation method on video retrieval performance". Proc. IEEE International Conference on Multimedia and Expo, pp. 5-8, Amsterdam, The Netherlands, 2005.

[3] O. Kalipzis, "Query processing in multimedia databases". Journal of Applied Science, Volume 2, pp. 109-113, 2002.

[4] H. Kosch and M. Döller, "Multimedia database systems: where are we now?" Institute of Information Technology, University Klagenfurt Universitätsstr. Austria, 2006.

[5] H. B. Kekre, "Image retrieval with shape features extracted using gradient operators and slope magnitude technique with BTC". International Journal of Computer Applications, Volume 6, Number 8, pp. 28-33, 2010.

[6] P. Sapra, S. Kumar, and R. K. Rathy, "Query processing in multilevel secure distributed databases". Proc. of the Fourth International Advance Computing Conference, 2014.

[7] P. Sapra, S. Kumar, R. K. Rathy, "Development of a concurrency control technique for multilevel secure databases". Proc. of the First International Conference on Reliability, Optimization and Information Technology, February 2014.

[8] X. Ma, D. Schonfeld and A. Khokhar, "Video event classification and image segmentation based on non-causal multi-dimensional hidden Markov models", IEEE Transactions on Image Processing, Vol. 18, No. 6, pp. 1304-1313, June 2009.

[9] B. V. Patel and B. B. Meshram, "Content based video retrieval systems". International Journal of UbiComp, Vol. 3, No. 2, pp. 13-30, 2012.

[10] T. C. Rakow, E. J., Neuhold and M. Lohr, "Multimedia database systems - the notions and the issues", Tagungsband GI-Fachtagung Datenbanksystems, Buro, Technik and Wissen-Schaft, Springer Informatik Aktuell, Berlin, 1995.

[11] C. Ribeiro and G. David, "A metadata model for multimedia databases". Proc. International Cultural Heritage Informatics Meeting, Archives and Museum Informatics, pp. 469-483, 2001.

[12] A. Burad, "Multimedia databases". Seminar Report, Roll No : 03005009, Computer Science and Engineering, Indian Institute of Technology, India, 2006.

[13] L. Lord and C. Pratt, "Retrievals from DB2 BLOB (Binary Large Objects) data warehouse using SAS". Proc. of NESUG Conference, 2000.

[14] D. Srivastava and Y. Velegrakis, "MMS: using queries as data values for metadata management". Proc. of the International Conference on Data Engineering, 2007.

[15] B. Panda, W. Perrizo, R. A. Haraty, "Secure transaction management and query processing in multilevel secure database systems". Proc. of the ACM Symposium on Applied Computing (ACM SAC 1994), pp. 363-368, 1994.

[16] R. A. Haraty and R. C. Fany, "Query acceleration in distributed database systems". Colombian Journal of Computation. Volume 2, Number 1, pp. 19-34, 2001.

[17] J. F. Abbass and R. A. Haraty, "Bit-level locking for concurrency control". Proc. of the Seventh ACS International Conference on Computer Systems and Applications (AICCSA 2009) – Sponsored by IEEE. Rabat, Morocco, pp. 168-173, May 2009.

 ISBN: 978-1-61208-415-2