Analytical Data Mining for Stream Data Analysis

PhD Progress Report, February 2006
SDDI 2006

Ronnie Alves (ronnie@di.uminho.pt)

Advisor: Orlando Belo (obelo@di.uminho.pt)

PhD Started at: February 2005.
PhD Ending (expected) at: February 2008.

Abstract. The main idea behind this research relies on analytical data mining functions to handle data streams. Given the characteristics of the data stream, the new methods and techniques for stream data analysis must conduct advanced analysis and data mining over fast and large data streams to capture the trends, patterns and exceptions. Besides, much of such data resides at rather low level of abstraction, whereas most analysts are interested in dynamic changes at relatively high levels of abstractions. Furthermore, recently studies are heading to combine ideas of *cube-based algorithms* with *data mining functions* to reveal exceptional and trend patterns over data streams. Thus, this work intends to provide new methods for effective and efficient analytical data mining over data streams.

1. Introduction

Previous studies (Babcock et al 2002, Garofalakis et al 2002) argue that mining data streams is challenging in the following two aspects. On the one hand, random access to fast and large data streams may be impossible. Thus multi-pass algorithms (i.e., ones that load data items into main memory multiple times) are often infeasible. On the other hand, the exact answers from data streams are often too expensive. Thus approximate answers are acceptable. While the above two issues are critical, they are not unique to data streams. For example, on-line mining very large databases also requires ideally one-pass algorithms and may also accept approximations.

Recent research indicates that a growing number of emerging applications, such as sensor networks, power supply, network traffic, stock exchange, telecommunications data flow and web click streams, have to handle various data streams. It is demanding to conduct advanced analysis and data mining over fast and large data streams to capture the trends, patterns and exceptions. Besides, much of such data resides at rather low level of abstraction, whereas most analysts are interested in dynamic changes at relatively high levels of abstractions. To discover such high level characteristics, one may need to perform *on-line multi-level analysis of stream* data, similar to on-line analytical processing (OLAP) of relational or multi-dimensional data (Dong et al 2002) (Chen et al 2002). Moreover, it must allow performing multiple data mining functions (such as frequent pattern analysis, classification, clustering, and so on).

The (on-line) *analytical data mining* paradigm was first introduced in (Han et al 1998, Jiawei Han 1998). They discussed the issues related to efficient and effective data mining in large data warehouses, including several aspects for integrating OLAP and data mining. The most important feature of such system is the *cube-based mining methods*.

High performance data cube technology is critical to analytical data mining. There have been many efficient data cube computation techniques developed in recent years (Dong et al 2006, Shao et al 2004, Feng et al 2004, Lakshmanan et al 2002).

Despite all those cubing algorithms, analytical data mining of stream data raises a basic issue (i.e., for data cube computing): *Given such characteristics of stream data, is it feasible to compute such data cube, since its size is usually much bigger than the original data set, and its construction may take multiple database scans?*

If it is possible to reach reasonable time, while computing such *stream data cube*, the next issue would be: *How to detect abnormal changes of cuboids cells, since on-line mining of the changes is one of the core issues in stream data analysis?* With data streams, people are more interested in mining queries like "*compared to the history, what are the distinct features of the current status?*" and "*what are the relatively factors over time?*"

2. The Problem

In this work we consider stream data as huge volume, infinite flow of data records, such as web click streams. The data is collected at the most detailed level in a multi-dimensional space, which may represent time, location, user, theme, and other semantic information.

To perform analytical data mining, we need to introduce the basic terms related to data cubes.

Let D be a relational table, called the base table, of a given cube. The set of all attributes A in D are partitioned into two subsets, the dimensional attributes DIM and the measure attributes M (then $DIM \cup M = A$ and $DIM \cap M = \varnothing$). The measure attributes functionally depend on the dimensional attributes in D and are defined in the context of the data cube using some typical aggregate functions, such as *COUNT*, *SUM*, *AVG*, *STD*, or some other statistical measure.

A tuple with schema A in a multi-dimensional space is called cell. Given three distinct cells *c1*,*c2* and *c3*, *c1* is an ancestor of *c2*, and *c2* a descendant of *c1* if on every dimensional attribute, either *c1* and *c2* share the same value, or *c1's* value is a generalized value of *c2's* in the dimension's concept hierarchy. *c2* is a sibling of *c3* if *c2* and *c3* have identical values in all dimensions except one dimension *A* where *c2(A)* and *c3(A)* have the same parent in the dimension's domain hierarchy. A cell which has *k* non-* values is called a k-d cell. (* indicates "all", the highest level on any dimension).

A tuple $c \in D$ is called base cell. A base cell does not have any descendant. A cell c is an aggregated cell if it is an ancestor of some base cell. For each aggregated cell c, its values on the measure attributes are derived from the complete set of descendant base cells of c.

Our task is to *perform high-level, analytical data mining over data streams in order to find unusual (exceptional) changes of trends, according to users' interest. In this sense computing (aggregating) interesting measures over stream cubes is a fundamental task.*

3. Our Research

In the beginning, we settled our efforts for exploring *frequent itemset mining*. Frequent itemset mining (FIM) is a core problem in many data mining tasks, and varied approaches to the problem appear in numerous papers across all data mining conferences. While the problem was introduced in the context of market basket analysis, the scope of the problem is much broader. As a fundamental operation in data mining, algorithms for FIM can be used as a building block for other, more sophisticated data mining processes.

Using the FIM ideas, we had implemented our itemset mining algorithm inspired on FP-Growth (Han et al 2000). It was proposed a pattern growth mining implementation which takes advantage of SQL-Extensions (we called *PGS*) on RDBMS systems for mining large transactional tables. Furthermore, RDBMS vendors offer a lot of features for taking control and management of the data. Integrating data mining in RDBMS is a quite promising area. There are several memory-approaches to mine all patterns. However, some few efforts have been made on database perspective, in those cases, only pure-SQL approaches. By using *PGS*, we can achieve competitive results and also avoid classical itemset mining bottlenecks: *candidate set generation and test (several expensive joins)*, and *table reconstruction*.

Next, we have experienced *PGS* combined with sequence mining in order to get *inter-transactional patterns*. The extracted patterns are augmented with two measures, the mean distance between the first and last itemsets of the pattern and the respective standard deviation of the distance. The introduction of a measure like the *coefficient of variation of distances* allows to easily rejecting patterns that show a significant distance variation and thus may represent misleading patterns.

We also have worked on the extraction of multi-dimensional association rules. It was started a project called MIUDA (*MIning Under supervision for real DAta applications*). MIUDA is a java application which takes advantage of data cube computing for getting multi-dimensional association rules over large data warehouses. Furthermore, it allows cube *processing incrementally* with (or without) concept hierarchies. This 1st version has implemented two algorithms: one for computing the cube (MOLAP-based with a Divide-and-Conquer strategy) and another for extracting its multi-dimensional rules (apriori-based).

We also have been working on two projects with industry since the 2nd quarter of 2005. One is related to telecommunications and other in retail. These projects help us to evaluate the knowledge base achieved during this research on real data applications, where the main task is to detect exceptional and interesting patterns.

Other interest application of FIM algorithms is on the computation of data cubes (since they have on their basis the lattice structure to exploit). In (Beyer & Ramakrishnan 1999) it was introduced an apriori-based approach for computing *iceberg data cubes*. Iceberg data cubes use an iceberg threshold condition (like minimum support on itemset mining or HAVING clause on SQL) to allow the computation of aggregated cells.

Our current research is focusing on iceberg cube mining with new pruning strategies. In fact, we exploit the previous issues and propose an efficient method for computing Maximal Correlated Cuboids Cell (called M3C-Cubing).

4. Our Agenda

In order to answer the first two questions mentioned on Section 1, we have decided to split our research activities on 5 steps. So, at the end of this research we intend to provide a complete (as possible) set of analytical data mining methods for stream data analysis.

We hope to submit for related conferences at least one report for each activity.

	1st Quarter	**2nd Quarter**	**3rd Quarter**	**4th Quarter**
2005	1a, 1b	1b, 1c	1c	1c, 2a, ~~2b~~, ~~2c~~
2006	2a, ~~2b~~, ~~2c, 2d~~	3a, 3b, 3c	4a, 4b, 4c	5a, 5b
2007	5b, 5c	5d	6	6
2008	Delivery!			

Note: the delayed activities are marked with a strikethrough line.

Activities

1) State of the art
    a. OLAP
    b. (click)Stream Analysis
    c. Data Mining
2) Computing data cubes
    a. State of the art
    b. Survey
    c. New methods
    d. Evaluation
3) Detecting abnormal/exceptional patterns
    a. State of the art
    b. New methods
    c. Evaluation
4) Detecting patterns changes over time
    a. State of the art
    b. New methods
    c. Evaluation
5) Analytical data mining
    a. State of the art
    b. Combine methods 2,3 and 4
    c. Evaluation
    d. Library
6) Thesis writing

5. The Current Research

The main idea of iceberg data cubing algorithms is to develop optimization techniques for computing only cuboids cells above certain minimum support (min_sup) threshold. Even using such techniques the curse of dimensionality remains, given the large number of cuboids to compute. Some efforts have been done on compressing those cuboids cells on *Closed cuboid*s. Nevertheless, for some of the dense databases, we consider in this work, even the set of all closed cuboids would grow to be too large. An alternative would be to compute only the *Maximal cuboids*. However, a pure maximal approaching implies loosing some information (i.e., we can generate the complete set of cuboids cells from its maximal but without their respective aggregation value).

To play with some loss information we need to add an interesting measure (we called *The Correlated Value of a Cuboid Cell*). This measure must disclose true correlation (also dependence) relationship among cuboids cells and has not to be influenced by the co-absence of cells pairs in the base relation table (i.e., needs to hold the *null-invariance property*). Furthermore, real world databases tend to be correlated, i.e., dimensions values are usually dependent on each other. In this work we exploit those issues and propose an efficient method for computing *Maximal Correlated Cuboids Cell* (called M3C-Cubing). Our performance study shows that our method is a promising candidate for scalable data cube computing.

6. Contributions

6.1 Publications

Programming Relational Databases for itemset Mining over Large Transactional Tables, with Orlando Belo. In Proceedings of 12th Portuguese Conference on Artificial Intelligence, **EPIA 2005**, LNAI Springer-Verlag, (Covilhã, Portugal, December, 2005).

> **Abstract**. Most of the itemset mining approaches are memory-like and run outside of the database. On the other hand, when we deal with data warehouse the size of tables is extremely huge for memory copy. In addition, using a pure SQL-like approach is quite inefficient. Actually, those implementations rarely take advantages of database programming. Furthermore, RDBMS vendors offer a lot of features for taking control and management of the data. We propose a pattern growth mining approach by means of database programming for finding all frequent itemsets. The main idea is to avoid one-at-a-time record retrieval from the database, saving both the copying and process context switching, expensive joins, and table reconstruction. The empirical evaluation of our approach shows that runs competitively with the most known itemset mining implementations based on SQL. Our performance evaluation was made with SQL Server 2000 (v.8) and T-SQL, throughout several synthetical datasets.

A Hybrid Method to Discover Inter-Transactional Rules, with Pedro Gabriel Ferreira, Paulo Azevedo, and Orlando Belo. In Proceedings of X Jornadas sobre Ingeniería del Software y Bases de Datos, **JISBD 2005**, (Granada, Spain, September, 2005).

> **Abstract**. Classical association rules are by nature intra-transaction based, i.e., the associations occurs among the item within the same transaction. Further, those

transactions are usually associated with a dimensional context which is typically ignored. This prevents that patterns, occurring along this dimension, are discovered. We introduced a method to for extraction and analysis of inter-transactional patterns. These patterns allow discovering relationships that occur among transactions within a dimensional frame. The method consists in the combination of association and sequence mining. First, association mining is applied to transform a dimensional transactional database into a sequence database. The latter is then mined in order to obtain frequent patterns. Sequence patterns are augmented with additional distance (among events) information. Finally, sequence rules are fulfilling user's constraints (with relation to the measures of confidence, lift and others) are extracted. These rules have applications in a wide range of domains like stock market, weather databases or in retail market pricing strategies.

6.2 Projects

MIUDA is acronym for MIning Under Supervision for Real Data Applications. Real Data Applications demands intensive operations for multi-dimensional on-line analysis over large databases. MIUDA is an analytical data mining tool which brings together concepts of OLAP and Data Mining. This 1st version provides mechanisms to extract *multi-dimensional association rules* from data cubes. (*OPIII Project* - LESI)

6.2 Workshop

We were in the program committee and co-editor of the (2nd - 2005) **Data GadGets Workshop** (in conjunction with JISBD Conference).

7. Related Work

There are several works related to the main topics of this research, but a few studies on analytical data mining techniques for handling data streams. It is presented here a short bibliography of the most significant efforts on this direction.

7.1 Stream Data Analysis

Recently, some interesting results have been reported for modeling and handling data streams (Babcock et al 2002), such as monitoring statistics over streams and query answering (Datar et al, Gehrke et 2001, Dobra et al 2002). Some previous work also involves change detection. For instance, the emerging patterns (Dong et al 1999) characterize changes from one data set to the other. In (Ganti et al 1999) some methods are proposed to measure the differences of the induced models in datasets. In (Domingos & Hulten 2000, Hulten et al 2001), the classification of time changing data streams is studied. Furthermore, conventional OLAP and data mining models have been extended to handle data streams, such as multi-dimensional analysis (Chen et al 2002), (Cai et al 2004) and clustering (Guha et al 2000).

7.2 Data Cube Computing

While the size complexity is a major issue of the data cube, to efficiently compute the data cube is another important research problem. Currently, most of the research efforts have been related to iceberg cubing (Beyer & Ramakrishnan 1999, Xin et al 2003,

Shao et al 2004) and closed/range cubing related algorithms (Lakshmanan et al 2002, Feng et al 2004, Dong et al 2006). Finally, stream data analysis using data cube was studied in (Cai et al 2004, Dong et al 2002, Chen et al 2002).

8. Final Discussion

In the first year we have been working on *frequent pattern mining* which gave us interesting ideas to explore on both data mining and data cube computing. They have on their basis methods and techniques for the best exploration of the itemset (or dimensions) *lattice*, and it is a fundamental operation in data mining. Besides, new researches on itemset mining such as *maximal* and *closed patterns* have been taken most attention on the last 7 years (from 1998 up to now). Further, there are some cubing-based algorithms exploring these compression techniques. Nevertheless, in terms of stream-based algorithms, may these cubing implementations suffer from the curse of dimensionality (*cube basis problem*). Therefore, it is important to complement these techniques with new *pruning strategies*. The new ideas must deal with *statistical aspects* of the data streams, *proper constraints*, *data mining and data cubing functions*, and also *tilted time window frame* through all the process.

9. References

A. Dobra, M. Garofalakis, J. Gehrke, and R. Rastogi. Processing complex aggregate queries over data streams. In SIGMOD'02, Madison, Wisconsin, June 2002.

B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In PODS'02, Madison, WI, June 2002.

D. Xin, J. Han, X. Li, and B. W. Wah, "Star-Cubing: Computing Iceberg Cubes by Top-Down and Bottom-Up Integration", Proc. 2003 Int. Conf. on Very Large Data Bases (VLDB'03), Berlin, Germany, Sept. 2003.

Dong Xin, Jiawei Han, Zheng Shao, and Hongyan Liu, "C-Cubing: Efficient Computation of Closed Cubes by Aggregation-Based Checking", in Proc. 2006 Int. Conf. on Data Engineering (ICDE'06), Atlanta, Georgia, April 2006.

G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In KDD'99, pages 43–52, San Diego, CA, Aug. 1999.

G. Dong, J. Han, L.V.S. Lakshmanan, J. Pei, H. Wang and P.S. Yu. "Online mining of changes from data streams: Research problems and preliminary results", In Proceedings of the 2003 ACM SIGMOD Workshop on Management and Processing of Data Streams. In cooperation with the 2003 ACM-SIGMOD International Conference on Management of Data (SIGMOD'03), San Diego, CA, June 8, 2003.

G. Hulten, L. Spencer, and P. Domingos. Mining time-changing data streams. In KDD'01, San Francisco, CA, Aug. 2001.

J. Gehrke, F. Korn, and D. Srivastava. On computing correlated aggregates over continuous data streams. In SIGMOD'01, pages 13–24, Santa Barbara, CA, May 2001.

J. Han, " Towards On-Line Analytical Mining in Large Databases", SIGMOD Record, 27(1):97-107, 1998.

J. Han, J. Pei, and Y. Yin, " Mining Frequent Patterns without Candidate Generation (PDF)", (Slides), Proc. 2000 ACM-SIGMOD Int. Conf. on Management of Data (SIGMOD'00), Dallas, TX, May 2000.

J. Han, S. Chee, and J. Y. Chiang, " Issues for On-Line Analytical Mining of Data Warehouses ", Proc. of 1998 SIGMOD'96 Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'98) , Seattle, Washington, June 1998, pp. 2:1-2:5.

K. Beyer and R. Ramakrishnan. Bottom-Up Computation of Sparse and Iceberg Cubes. SIGMOD'99.

L. V. S. Lakshmanan, J. Pei, and J. Han, " Quotient Cube: How to Summarize the Semantics of a Data Cube ", Proc. 2002 Int. Conf. on Very Large Data Bases (VLDB'02), Hong Kong, China, Aug. 2002.

M. Datar, A. Gionis, P. Indyk, and R. Motwani. Maintaining stream statistics over sliding windows (extended abstract). citeseer.nj.nec.com/491746.html.

M. Garofalakis, J. Gehrke, and R. Rastogi. Querying and mining data streams: You only get one look. In VLDB’02, Hong Kong, China, Aug. 2002.

P. Domingos and G. Hulten. Mining high-speed data streams. In KDD’00, pages 71–80, Boston, MA, Aug. 2000.

S. Guha, N. Mishra, R. Motwani, and L. O’Callaghan. Clustering data streams. In FOCS’00, pages 359–366, Redondo Beach, CA, 2000.

V. Ganti, J. Gehrke, and R. Ramakrishnan. A framework for measuring changes in data characteristics. In PODS’99, pages 126–137, Philadelphia, PA, May/June 1999.

Y. Chen, G. Dong, J. Han, B. W. Wah, and J. Wang. Multi-dimensional regression analysis of time-series datastreams. In VLDB’02, Hong Kong, China, Aug. 2002.

Y. Chen, G. Dong, J. Han, J. Pei, B. W. Wah, and J. Wang, " Online Analytical Processing Stream Data: Is It Feasible? ", Proc. 2002 ACM-SIGMOD Int. Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'02), Madison, WI, June 2002.

Y. D. Cai, D. Clutter, G. Pape, J. Han, M. Welge, and L. Auvil, “MAIDS: Mining Alarming Incidents from Data Streams”, (system demonstration), Proc. 2004 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'04), Paris, France, June 2004.

Ying Feng, Divyakant Agrawal, Amr El Abbadi, Ahmed Metwally: Range CUBE: Efficient Cube Computation by Exploiting Data Correlation. ICDE 2004: 658-670

Z. Shao, J. Han, and D. Xin, “MM-Cubing: Computing Iceberg Cubes by Factorizing the Lattice Space”, Proc. 2004 Int. Conf. on Scientific and Statistical Database Management (SSDBM'04), Santorini Island, Greece, June 2004.