CS 641: Multimedia Database Systems

Midterm Take-Home Exam

Mohamed G. Elfeky
mgelfeky@cs.purdue.edu

October 28, 2002

Mining Data Streams

1 Introduction

Data mining is the process of discovering patterns and trends by sifting through large amounts of data using technology that employs statistical and mathematical techniques. Started at the early 90's for transactional databases [1], data mining research has emerged significantly during one decade. All types of media databases have been investigated in data mining research. To mention a few, image databases [16], spatial databases [8], video databases [15], and audio databases [14]. As an emerging new type of data that involves media types, data streams impose several challenges to data mining research.

Real-time systems and dynamic environments often generate tremendous, potentially infinite, amount of stream data [5]. Life embraces several examples for data streams, e.g., meteorological data containing several weather measurements (temperature, humidity, etc.), telecommunication networks recording telephone calls and Internet flow traces, media networks producing lots of multimedia (image, audio, and video) streams, etc. As the volume of data streams is too huge to be stored on disks or scanned multiple times, the need for online data mining techniques is inevitable.

The structure of the rest of this report is as follows. Section 2 presents the features of data streams that led to data mining challenges along with a brief literature review. Section 3 gives some practical examples of applications that would benefit from mining data streams research. Section 4 summarizes one of the research efforts done in mining data streams. Section 5 serves as a roadmap of the future work that should be done in mining data streams. In the appendix, a specific related paper is reviewed.

2 Research Issues

Traditional database systems have been oriented toward business data processing, and consequently are designed to address the needs of these applications. Data mining techniques, in turn, has been designed in a similar fashion. Data streams, however, differ substantially from traditional databases. Differences are illustrated in [3]. Of particular interest to data mining research are two differences. First, while traditional databases assume that data elements are synchronized and so exact mining results can always be obtained, streams data is often lost, stale, or intentionally omitted for processing reasons and so mining results must be computed with incomplete information. Second, traditional databases assume that applications require no real-time services whereas data streams are real-time by nature; requiring online data mining techniques.

Therefore, the challenging task of mining data streams is to develop online incremental data mining models that operate on incomplete data, or alternatively, data summaries, yet that can obtain "good" results.

Incremental mining is defined as how to maintain the discovered patterns over time as data is continuously being added to the database, without repeating the whole mining process. The term is originally proposed by Agrawal and Psaila in [2]. Incremental mining has been investigated in various types of mining patterns such as association rules [17], clustering patterns [9], decision trees [18], sequential patterns [19], etc. Online mining considers providing the user with the ability to change the mining parameters while the mining process is in progress. Generally, an online mining algorithm outputs continuous results as long as the user keeps changing the parameters. Online mining has been investigated for association rules [13], and for time sequences [20]. Mining on incomplete data or data summaries has not been investigated since it was not required as discussed before. However, in this context, data mining would benefit from work on data summaries such as histograms [4], wavelets [11] and correlated aggregates [10].

Research in mining data streams is very recent, and so a little literature can be found. A domain-specific language, Hancock [6], has been designed at AT&T to extract signatures from massive transaction streams. This work will be reviewed in the appendix. Algorithms for constructing decision trees [7] and clustering [12]. In [5], Chen et al. investigated regression analysis methods for summarizing data streams and building online data cubes.

3 Practical Applications

Applications where mining data streams would be in practical use include military applications, financial market, telecommunication networks, industrial corporations, etc.

In a military application, sensors worn by soldiers will continuously read and send several measurements, e.g., blood pressure, heart rate, and geographical position. A data mining application to these streams is to discover associations, if any, between the geographical position and the heart rate, and therefore the top leaders can take decisions of the best positions to camp the soldiers.

Financial market analysis is always one of the most benefited applications from data mining. Analyzing the periodicity of stock prices monitored every minute, and discovering the periodic patterns, if any, of those prices, would let brokers to know better when to buy or to sell a specific stock. Moreover, associating this periodicity with the internal decisions of the company would let the decision makers know when it is better to take a specific decision so that their stock price get higher.

As one of the largest telecommunication companies in USA, AT&T is collecting approximately 300M call logs per day from 100M customers. Analyzing the call logs with data mining techniques may result in providing more profitable call plans to certain customers. Monitoring Internet log streams leads to developing more powerful intrusion detection techniques. Knowing that intrusion detection is mainly based on data mining techniques proves the broad benefit of mining data streams.

There are so many applications for industrial corporations. One of them, taken from [5], is as follows. A power supply station collects streams of power usage data, with the lowest granularity as user, location, and minute. Linear regression analysis is a common data mining technique for discovering dominant trends of such time data. Given a large number of users, it is only realistic to analyze the fluctuation of power usage at certain high levels, such as by city or district and by hour, making timely power supply adjustments and handling unusual situations. Data cubes are used by data mining techniques for this type of applications to allow mining on the lower levels as well as the higher ones. Therefore, regression analysis of data streams stored in data cubes is a practical research issue that has been studied in [5], and will be briefed in the next section.

4 Regression Analysis of Time Series Data Streams

In this work, the authors propose an online data mining technique for time series data streams. The data mining technique proposed is based on using regression analysis as a way for data summarization, and on using data cubes as a materialization view of the data in order to be able to perform the mining and introduce the results on multiple levels.

The authors proved mathematically that for linear regression analysis, only a small number of regression measures rather than the complete stream of data need to be used. As the space and time are critical for online computation, a partially materialized data cube will be computed rather than the fully materialized one. The authors proved that this partial model is sufficient for most analysis tasks, and at the same time ensures that the total amount of data to retain in memory is small. Moreover, not all the data cube will be stored in memory. Only two layers, selected according to the application, will be stored while the other layers can be computed online from these two layers. The authors proposed two alternative methods for computing those other layers, and compared them empirically.

An extensive performance study has been conducted to study the proposed approach. Synthetic datasets with 100,000 data streams have been used. Experiments were performed on a 750MHz AMD PC with 512M bytes main memory, running Microsoft Windows-2000 Server. The experiments examine the performance of time and space with respect to the percentage of aggregated computed cells of the cube, the size of the selected layers to be stores, and the number of dimensions and levels in the data. The study shows that the approach is time and space efficient for the studied factors, and the results are of practical interest.

5 Future Roadmap

The work introduced in the previous section can be extended in so many ways. Other types of data summaries can be studied with data cubes. To my knowledge, there is no research has been done to study how to implement different types of data summaries into data cubes. To study one of those types, one should study thoroughly its features, and try to examine the compatibility between those features and data cube. For example, studying histograms, one should show that computing some specific histogram cube cells would lead to the ease of computation of the other cube cells.

References

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining Association Rules between Sets of Items in Large Databases. In *Proc. of the 1993 ACM SIGMOD Int. Conf. on Management of Data*, Washington, D.C., May 1993.

[2] R. Agrawal and G. Psaila. Active Data Mining. In *Proc. of the 1st Int. Conf. on Knowledge Discovery and Data Mining*, Montreal, Canada, August 1995.

[3] D. Carney, U. etintemel, M. Cherniack, C. Convey, S. Lee, G. Seidman, M. Stonebraker, N. Tatbul, and S. Zdonik. Monitoring Streams - A New Class of Data Management Applications. In *Proc. of the 28th Int. Conf. on Very Large Data Bases*, Hong Kong, China, August 2002.

[4] S. Chaudhuri, R. Motwani, V. Narasayya. Random Sampling for Histogram Construction: How much is enough? In *Proc. of the 1998 ACM SIGMOD Int. Conf. on Management of Data*, Seattle, Washington, June 1998.

[5] Y. Chen, G. Dong, J. Han, B. Wah, and J. Wang. Multi-Dimensional Regression Analysis of Time-Series Data Streams. In *Proc. of the 28th Int. Conf. on Very Large Data Bases*, Hong Kong, China, August 2002.

[6] C. Cortes, K. Fisher, D. Pregibon, and A. Rogers. Hancock: A Language for Extracting Signatures from Data Streams. In *Proc. of the 6th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Boston, Massachusetts, August 2000.

[7] P. Domingos and G. Hulten. Mining High-Speed Data Streams. In *Proc. of the 6th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Boston, Massachusetts, August 2000.

[8] M. Ester, H. Kriegel, and J. Sander. Knowledge Discovery in Spatial Databases. In *Proc. of the 23rd Annual German Conf. on Artificial Intelligence*, Bonn, Germany, September 1999.

[9] M. Ester, H. Kriegel, J. Sander, M. Wimmer, and X. Xu. Incremental Clustering for Mining in a Data Warehousing Environment. In *Proc. of the 24th Int. Conf. on Very Large Data-bases*, New York City, New York, August 1998.

[10] J. Gehrke, F. Korn, and D. Srivastava. On Computing Correlated Aggregates over Continual Data Streams. In *Proc. of the 2001 ACM SIGMOD Int. Conf. on Management of Data*, Santa Barbara, California, May 2001.

[11] A. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. Strauss. Surfing Wavelets on Streams: One-Pass Summaries for Approximate Aggregate Queries. In *Proc. of the 27th Int. Conf. on Very Large Data Bases*, Roma, Italy, September 2001.

[12] S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan. Clustering Data Streams. In *Proc. of the 41st Annual Symposium on Foundations of Computer Science*, Redondo Beach, California, November 2000.

[13] C. Hidber. Online Association Rule Mining. In *Proc. of the 1999 ACM SIGMOD Int. Conf. on Management of Data*, Philadelphia, Pennsylvania, June 1999.

[14] B. Matichuk and O. Zaane. Unsupervised Classification of Sound for Multimedia Indexing. In *Proc. of the 1st Int. Workshop on Multimedia Data Mining*, Boston, Massachusetts, August 2000.

[15] J. Pan and C. Faloutsos. VideoGraph: A New Tool for Video Mining and Classification. In *Proc. of the ACM/IEEE Joint Conference on Digital Libraries*, Roanoke, Virginia, June 2001.

[16] J. Roden, M. Burl, and C. Fowlkes. The Diamond Eye Image Mining System. In *Proc. of the 11th Int. Conf. on Scientific and Statistical Database Management*, Cleveland, Ohio, July 1999.

[17] S. Thomas, S. Bodagala, K. Alsabti, and S. Ranka. An Efficient Algorithm for the Incremental Updation of Association Rules in Large Databases. In *Proc. of the 3rd Int. Conf. on Knowledge Discovery and Data Mining*, New Port Beach, California, August 1997.

[18] P. Utgoff. ID5: An Incremental ID3. In *Proc. of the 5th Int. Conf. on Machine Learning*, pages 107-120, Morgan Kaufmann, 1988.

[19] K. Wang and J. Tan. Incremental Discovery of Sequential Patterns. In *Proc. of the SIGMOD Data Mining Workshop on Research Issues on Data Mining and Knowledge Discovery*, Montreal, Canada, May 1996.

[20] B. Yi, N. Sidiropoulos, T. Johnson, H. Jagadish, C. Faloutsos, A. Biliris. Online Data Mining for Co-Evolving Time Sequences. In *Proc. of the 16th Int. Conf. on Data Engineering*, San Diego, California, February 2000.

Appendix: Paper Review

C. Cortes, K. Fisher, D. Pregibon, and A. Rogers. Hancock: A Language for Extracting Signatures from Data Streams. In *Proc. of the 6th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Boston, Massachusetts, August 2000.

Note: This paper has received the best research paper award.

Summary

The paper presents a novel idea of a domain-specific programming language based on C that is used for writing efficient yet readable data streams programs.

Although a preliminary version of the paper appeared before, the language presented earlier was for only one particular form of data. In this paper, the language is generalized for arbitrary data streams. However, the paper lacks examples other than the "Cell Tower" example. Such other examples would clarify the significance of this language to the general view of data streams.

Comments

Main Contributions

- Given that the language is presented in an earlier version for a specific type of data streams, the main contribution in the paper is the generalization they discuss for all data streams.

- In spite of the previous fact, the discussion about the generalization is not clear through the lines (it can be inferred by the careful reader) nor are there examples to clarify this generalization.

Positive Aspects

- The idea is novel and the language is presented in a clear way (see the first negative aspect).

- The authors discuss clearly why they chose a language rather than a library.

- Section 5 acts as a case study using the language in a real system.

Negative Aspects

- The paper is about a new language. The clauses of the language are presented in a "reason-driven" way. In other words "because we need this in the running example of data streams, Hancock has that clause". Although this is a clear way of presenting a language, the running example itself is not clear or more specifically, not all its components and its needs are clear. For example, the idea of having hierarchical event structure is not clear. Is it like an event will trigger the other or what? Therefore, the idea of having it as a multi-union type is not clear.

- Although the paper presentation depends greatly on the running example in Section 2, Section 2 is very small and not describing the problem well. It would be better if the section includes a well introduction about the problem along with the programming issues required for it instead of having these issues scattered in many subsections in Section 3. In other words, Section 2 should have been self-contained.

- In each subsection of Section 3, a simple example for using the language in the Cell Tower example is given. A hint is given about the general case without examples despite the fact that the paper should be for the general case not only for this specific running example.

- In Section 3.4,

  - The operator :+: defining the union of two multi-unions is not well defined: the case in which the two multi-unions have the same label with different values is not discussed although it may be inferred, intuitively, as the same event happened twice, yet what if these two labels are not events.
  - The use of multi-unions as labels not events is not discussed although it is the general case of multi-unions.