Dependable Real-time Data Mining

Bhavani Thuraisingham, [1, 2] Latifur Khan [1], Chris Clifton [2, 3],
John Maurer [2], Marion Ceruti [4]

1. *The University of Texas at Dallas*
2. *The MITRE Corporation*
3. *Purdue University*
4. *Space and Naval Warfare Systems Center, San Diego Code 24121,*

Abstract

In this paper we discuss the need for real-time data mining for many applications in government and industry and describe resulting research issues. We also discuss dependability issues including incorporating security, integrity, timeliness and fault tolerance into data mining. Several different data mining outcomes are described with regard to their implementation in a real-time environment. These outcomes include clustering, association-rule mining, link analysis and anomaly detection. The paper describes how they would be used together in various parallel-processing architectures. Stream mining is discussed with respect to the challenges of performing data mining on stream data from sensors. The paper concludes with a summary and discussion of directions in this emerging area.

1. Introduction

Data mining is the process of posing various queries and extracting useful and often previously unknown and unexpected information, patterns, and trends from large quantities of data, generally stored in databases. These data could be accumulated over a long period of time or they could be large data sets accumulated simultaneously from heterogeneous sources such as different sensor types. The goals of data mining include improving marketing capabilities, detecting abnormal patterns, and predicting the future based on past experiences and current trends. There is clearly a need for this technology for many applications in government and industry. For example, a marketing organization may need to determine who their potential customers are. There are large amounts of current and historical data being stored. Therefore, as databases become larger, it becomes increasingly difficult to support decision making. In addition, the data could be from multiple sources and multiple domains. There is a clear need to analyze the data to support planning and other functions of an enterprise.

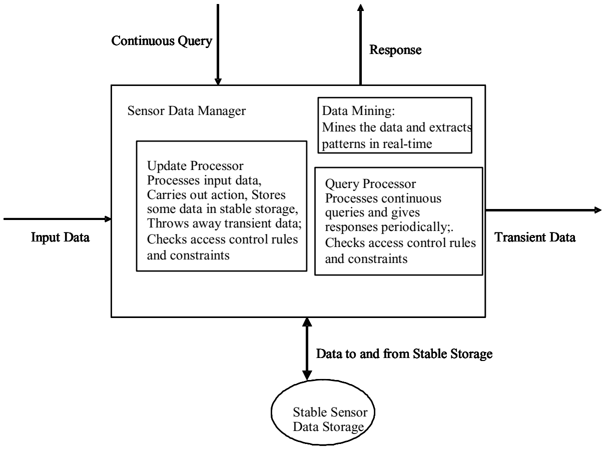
Continuous Query
Response
Sensor Data Manager
Data Mining:
Mines the data and extracts patterns in real-time
Update Processor
Processes input data,
Carries out action, Stores some data in stable storage,
Throws away transient data;
Checks access control rules and constraints
Query Processor
Processes continuous queries and gives responses periodically;.
Checks access control rules and constraints
Input Data
Transient Data
Data to and from Stable Storage
Stable Sensor Data Storage

Figure 1. Concept of Operation for Real-time Data Management and Data Mining

Much of the focus on data mining has been for analytical applications. However there is a clear need to mine data for applications that have to meet timing constraints. For example, a government agency may need to determine whether a terrorist activity will happen within a certain time or a financial institution may need to give out financial quotes and estimates within a certain time. That is, we need tools and techniques for real-time data mining. Consider for example a medical application where the surgeons and radiologists have to work together during an operation. Here, the radiologist has to analyze the images in real-

Proceedings of the Eighth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC'05)
0-7695-2356-0/05 $ 20.00 **IEEE**

time and give inputs to the surgeon. In the case of military applications, images and video may arrive from the war zone. These images have to be analyzed in real-time so that advice is given to the war fighter. The challenge is to determine which data to analyze and which data to discard for future analysis in non real-time. In the case of counter-terrorism applications, the system has to analyze the data about the passenger from the time the passenger gets ticketed until the plane is boarded, and give proper advice to the security agent. For all of these applications there is an urgent need for real-time data mining. Figure 1 illustrates a concept of operation for real-time data management and mining where some data are discarded, other data are analyzed and a third data set is stored for future use.

Thuraisingham, et al. introduced the notion of real-time data mining in [TCCM]. In that paper we focused on mining multimedia data, which is an aspect of real-time data mining. Since then there have been many developments in sensor data management as well as stream data mining. Furthermore, the need for real-time data mining is more apparent especially due to the need for counter-terrorism applications. In this paper we continue with out investigation of real-time data mining issues and challenges.

The organization of this paper is as follows. Some issues in real-time data mining including a discussion of real-time threats are given in Section 2. Adapting data mining techniques to meet real-time constraints is described in Section 3. Parallel and distributed real-time data mining is discussed in section 4. Techniques in dependable data mining that integrate security real-time processing and fault tolerance are given in Section 5. Stream data mining is discussed in Section 6. Summary and directions are provided in Section 7.

2. Issues in Real-time Data Mining

As stated in Section 1, data mining has typically been applied to non-real-time analytical applications. Many applications, especially for counter-terrorism and national security, need to handle real-time threats. Timing constraints characterize real-time threats. That is, such threats may occur within a certain time and therefore we need to respond to them immediately. Examples of such threats include the spread of smallpox virus, chemical attacks, nuclear attacks, network intrusions, and bombing of a building before. The question is, what types of data mining techniques do we need for real-time threats?

Data mining can be applied to data accumulated over a period of time. The goal is to analyze the data, make deductions and predict future trends. Ideally it is used as a decision support tool. However, the real-time situation is entirely different. We need to rethink the way we do data mining so that the tools can produce results in real-time.

For data mining to work effectively, we need many examples and patterns. We observe known patterns and historical data and then make predictions. Often for real-time data mining as well as terrorist attacks we have no prior knowledge. So the question is, how do we train the data mining tools based on, say, neural networks without historical data? Here we need to use hypothetical data as well as simulated data. We need to work with counter-terrorism specialists and get as many examples as possible. When we have gathered the examples and start training the neural networks and other data mining tools, the question becomes what sort of models do we build? Often the models for data mining are built before hand. These models are not dynamic. To handle real-time threats, we need the models to change dynamically. This is a big challenge.

Data gathering is also a challenge for real-time data mining. In the case of non real-time data mining, we can collect data, clean data, format the data, build warehouses and then carry out mining. All these tasks may not be possible for real-time data mining due to time constraints. Therefore, the questions are what tasks are critical and what tasks are not? Do we have time to analyze the data? Which data do we discard? How do we build profiles of terrorists for real-time data mining? How can we increase processing speed and overall efficiency? We need real-time data management capabilities for real-time data mining.

From the pervious discussion it is clear that a lot has to be done before we can perform real-time data mining. Some have argued that there is no such thing as real-time data mining and it will be impossible to build models in real-time. Some others have argued that without accurate data we cannot do effective data mining. These arguments may be true. However, others have predicted the impossibility of technology (e.g. air travel, internet) that today we take for Granted. Our challenge is to then perhaps redefine data mining and figure out ways to handle real-time threats.

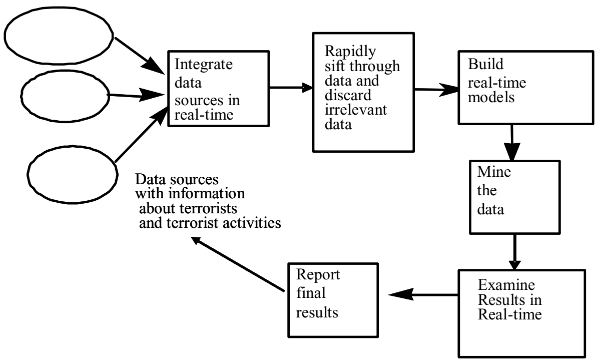
Integrate data sources in real-time
Rapidly sift through data and discard irrelevant data
Build real-time models
Mine the data
Examine Results in Real-time
Report final results
Data sources with information about terrorists and terrorist activities

Figure 2. Real-time Data Mining Cycle

As we have stated, there are several situations that have to be managed in real-time. Examples are the spread of smallpox, network intrusions, and analyzing data sensor data. For example, surveillance cameras are placed in various places such as shopping centers and in front of embassies and other public places. Often the data from these sensors must be analyzed in real-time to detect/prevent attacks. We discuss some of the research directions in the remaining sections. Figure 2 illustrates the cycle for real-time data mining.

3. Real-time Data-Mining Techniques

In this section we examine the various data mining outcomes and discuss how they could be applied for real-time applications. The outcomes include making associations, link analysis, cluster formation, classification and anomaly detection. The techniques that result in these outcomes are based on neural networks, decisions trees, market basket analysis techniques, inductive logic programming, rough sets, link analysis based on the graph theory, and nearest neighbor techniques. As we have stated in [THUR03], the methods used for data mining are top-down reasoning where we start with a hypothesis and then determine whether the hypothesis is true or bottom-up reasoning where we start with examples and then form a hypothesis.

Let us start with association mining techniques. Examples of these techniques include market basket analysis techniques [AIS93]. The goal is to find which items go together. For example, we may apply a data-mining tool to a data set and find that John comes from Country X and he has associated with James who has a criminal record. The tool also outputs the result that an unusually large percentage of people from Country X have performed some form of terrorist attacks. Because of the associations between John and Country X, as well as between John and James, and James and criminal records, one may conclude that John has to be under observation. This is an example of an association. Link analysis is closely associated with making associations. Whereas association-rule based techniques are essentially intelligent search techniques, link analysis uses graph theoretic methods for detecting patterns. With graphs (i.e., nodes and links), one can follow the chain and find links. For example A is seen with B and B is friends with C and C and D travel a lot together and D has a criminal record. The question is what conclusions can we draw about A? Now, for real-time applications we need association-rule mining and link analysis techniques that out put the associations and links in real-time.

Relevant research is in progress. Incremental association rule mining techniques were first proposed in [CHNW96]. More recently, data-stream techniques for mining association have been proposed [CWYM04]; these will be discussed further in Section 6. Whereas they address some of the issues faced by real-time data mining, the key issue of time-critical need for results has not been addressed. The real-time database researchers have developed various techniques including real-time scheduling and approximate-query processing. We need to examine similar techniques for association-rule mining and link analysis and determine the outcomes that can be determined in real time. Are we losing information by imposing real-time constraints? How can we minimize errors when we impose real-time constraints? Are approximate answers accurate enough to base decisions on them?

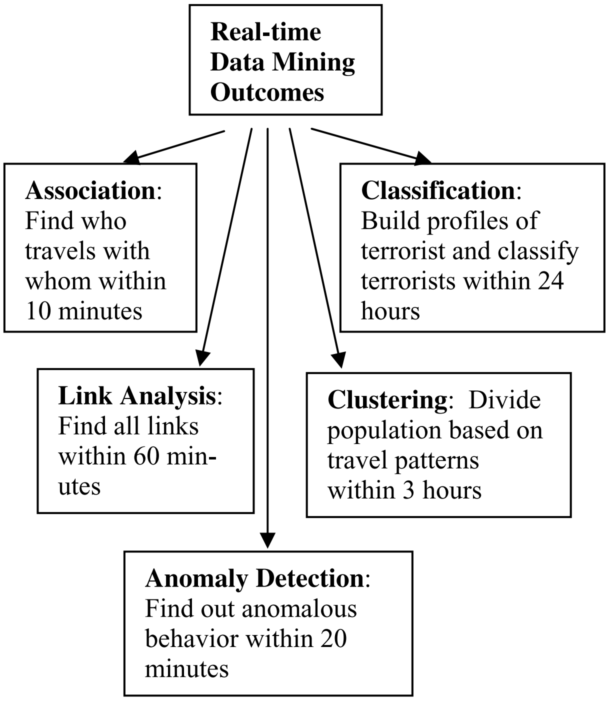
Real-time Data Mining Outcomes
Association: Find who travels with whom within 10 minutes
Classification: Build profiles of terrorist and classify terrorists within 24 hours
Link Analysis: Find all links within 60 min-utes
Clustering: Divide population based on travel patterns within 3 hours
Anomaly Detection: Find out anomalous behavior within 20 minutes

Figure 3. Real-time Data Mining Outcomes

Next, let us consider clustering techniques. One could analyze the data and form various clusters. For example, people with origins from country X and who belong to a certain religion may be grouped into Cluster I. People with origins from country Y and who are less than 50 years old may form another Cluster II. These clusters could be formed based on their travel patterns, eating patterns, buying patterns, or behavior patterns. Whereas clustering techniques do not rely on any pre-specified condition to divide the population, classification divides the population based on some predefined condition. The condition is found based on examples. For example, we can form a profile of a terrorist. He could have the following characteristics: Male less than 30 years of a certain religion and of a certain ethnic origin. This means all males less than 30 years belonging to the same religion and the same ethnic origin will be classified into this group and possibly could be placed under observation. The examples of clustering and classification discussed above are for analytical applications. For real-time applications the challenges is to find the important clusters in real-time. Again, data stream techniques may provide a start. Another approach is iterative techniques. Classical clustering methods such as *k*-means and EM could refine answers based on the time available rather than terminating on distance-based criteria. The question is, how much accuracy and precision are we sacrificing by imposing timing constraints?

Another data mining outcome is anomaly detection. A good example here is learning to fly an airplane without wanting to learn to takeoff or land. The general pattern is that people want to get a complete training course in flying. However there are now some individuals who want to learn flying but do not care about take off or landing. This is an anomaly. Another example is John always goes to the grocery store on Saturdays. But on Saturday October 26, 2002 he goes to a firearms store and buys a rifle. This is an anomaly and may need some further analysis as to why he is going to a firearms store when he has never done so before. Is it because he is nervous after hearing about the sniper shootings or is it because he has some ulterior motive? If he is living, say, in the Washington DC area, then one could understand why he wants to buy a firearm, possibly to protect him. But if he is living in say Socorro, New Mexico, then his actions may have to be followed up further. Anomaly detection faces many challenges even ignoring time constraints; a prime example is approaches for Intrusion Detection (see [LF01] and [AX99] for surveys of the problem, and [WS04] for a recent discussion of anomaly detection approaches.) Adding real-time constraints will only exacerbate the difficulties. In many cases the anomalies have to be detected in real-time both for cyber security as well as for physical security. The technical challenge is to come up with meaningful anomalies as well as meet the timing constraints; however, a larger issue is to define the problems and surrounding systems to take advantage of anomaly detection methods in spite of the (inevitable?) false positives and false negatives. Figure 3 illustrates examples of real-time data mining outcomes.

4. Parallel, Distributed, Real-Time Data Mining

For real-time data-mining applications, perhaps a combination of techniques may prove most efficient. For example, association-rule techniques could be applied either in series or in parallel with clustering techniques, which is illustrated in Figure 4. In series, the association-rule technique may provide enough information to issue a real-time alert to a decision maker before having to invoke the clustering algorithms.

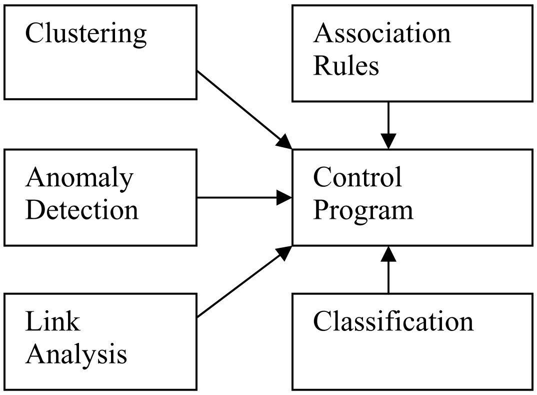

Figure 4. Data-mining tasks executing in concert on separate platforms with direct link to the control program

By using parallel processing software that executes on one or more hardware platforms with multiple processors, several real-time data-mining techniques can be explored simultaneously rather than sequentially. Among the many ways to implement this, two basic categories emerge: First, one can execute real-time data-mining programs simultaneously but on separate processors and input the results to a control program that compares the results to criteria or threshold values to issue alert reports to a decision maker.

The second category is an architecture in which the programs execute in parallel, either on the same hardware platform or over a network, as depicted in Figure 5, where a central program would format and parse data inputs to the various processors running the programs to determine the different data-mining outcomes. For example, when clusters start to form in the output of the cluster-detection processor, these clusters could be compared to the associations found in the association-rule processor. Similarly, the patterns formed by the link analysis processor could be input into the anomaly detector for examination to see if the pattern is the same or different from those expected. The various processors could all process the same data in different ways, or they could process data from different sources. The central control program could compare the results to the criteria or thresholds and issue alerts even before the slower algorithms have finished processing. The control program would continue to send newly emerging results to the decision maker while the response to the threat is in progress.

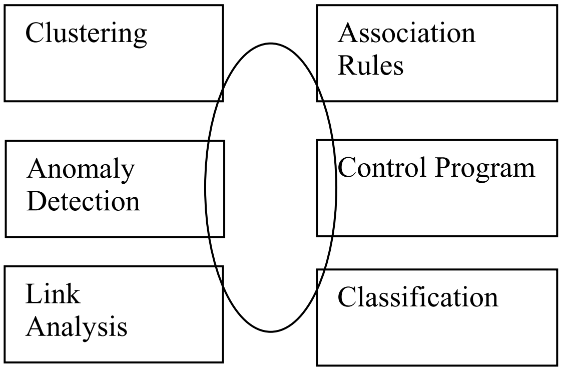

Figure 5. Distributed data-mining tasks executing on a network

The method of parallel processing depicted in Figure 6 is potentially the fastest and most efficient method of real-time data mining because the software that implements every data-mining outcome executes on the same hardware platform without any of the delays associated with communications networks, routers, etc. It is also the most versatile, challenging and potentially best implement using artificial intelligence (AI) techniques for pattern recognition and algorithm coordination. These AI techniques could include rule-based reasoning, case-based reasoning, and Bayesian networks.

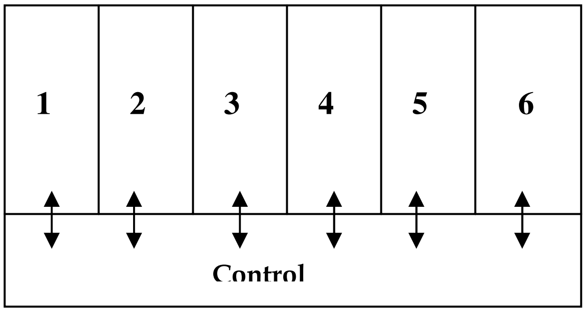

Figure 6. Data-mining tasks executing on a parallel machine

5. Dependable Data Mining

For a system to be dependable it must be secure, fault tolerant, meet timing deadlines, and manage high quality data. However, integrating these features into a system means that the system has to meet conflicting requirements determined by the policy makers and the applications specialists. For example, if the systems make all the access control checks, then it may miss some of its deadlines. The challenge in designing dependable systems is to design systems that are flexible. For example, in some situations it may be important to meet all the timing constraints while in some other situations it may be critical to satisfy all the security constraints.

The major components of dependable systems include dependable networks, dependable middleware including infrastructures, dependable operating systems, dependable data managers and dependable applications. Data mining, which can be regarded as an aspect of information and data management, has to be dependable also. This means that the data mining algorithms have to have the ability to recover from faults as well as maintain security, and meet real-time constraints all in the same program.

Sensor data may be available the form of streams. Special data-management systems are needed to process stream data. For example, much of the data may be transient data. Therefore, the system has to analyze the data, discard unneeded data, and store the necessary data all in real time. Special query processing strategies including query optimization techniques are needed for data-stream management. Many of the queries on stream data are continuous queries.

Aggregating making sense out of the sensor data is a major research challenge. The data may be incomplete or sometimes inaccurate. Many data prove to be irrelevant, thus increasing the noise of the detection-in-clutter task. We need the capability to deal with uncertainty and reason with incomplete data. Information management includes extracting knowledge and models from data as well as mining and visualizing the data. Much work has been accomplished in information management the last several years. For example, sensor data must be visualized for a better understanding of the data. We need to develop intelligent, real-time visualization tools for the sensor data. One may also need to aggregate the sensor data and possibly build repositories and warehouses. However, much of the sensor data may be transient. Therefore, we need to determine which data to store and which data to discard. Data may also have to be processed in real-time. Some of the data may be stored and possibly warehoused and analyzed for conducting analysis and predicting trends. That is, the sensor data from surveillance cameras must be processed within a certain time. The data may also be warehoused for subsequent analysis.

Sensor and stream data mining are becoming an important areas. We need to examine the data-mining techniques such as association rule mining, clustering and link analysis for sensor data and data streams from sensors and other devices. One important consideration is to select the level of granularity at which to mine the data. For example, should we mine raw sensor data or data processed at a higher level of aggregation? For example, patterns found in images are easily detected only when observed in the image context where the relationships between image features are preserved. These features are not recognized easily by analyzing a series of pixels from the image.

As we have stressed, we need to manage sensor data in real-time. Therefore, we may need to mine the data in real-time also. This means not only building models ahead of time so that we can analyze the data in real-time, we may also need to build models in real-time. That is, the models have to be flexible and dynamic. Model formation represents the aggregation of information at a very high level. This is a major challenge. As we have stated in section 2, we also need many training examples to build models. For example, we need to mine sensor data to detect, and possibly prevent, terrorist attacks. This means that we need training examples to train the neural networks, classifiers and other tools so that they can recognize in real-time when a potential anomaly occurs. Sensor-data mining is a fairly new research area and we need a research program for sensor-data management and data mining. The mining of data streams is discussed in Section 6.

Data mining may be a solution to some dependability issues. Most data mining techniques generate aggregates over large quantities of data; averaging out random errors. Systematic errors pose a greater challenge, but as shown in [AS00] and randomization approaches to privacy-preserving data mining, knowing something about the source of errors allows high quality data mining even if we cannot reconstruct correct data. Many techniques are non-deterministic; the similarity or dissimilarity of results of repeated runs provides a measure of dependability. (This was used to minimize errors in anomaly detection in [CL03].) Data mining has the potential to improve the dependability of decisions based on data, even if each datum taken separately is not dependable.

Data mining has also come under fire, perhaps unfairly, because of perceived impacts on privacy. Researchers are developing techniques for privacy-preserving data mining as well as for s handling the inference problem that occurs through data mining [VC04]. Many real-time data mining problems involve sensitive data; privacy will remain an issue. Many privacy-preserving data mining approaches come at a significant computational cost; we need to integrate security techniques with real-time data mining so that we can develop algorithms for dependable data mining. In particular, methods that trade off security for time constraints may be appropriate for particular problems. A passenger trying to catch a plane may be willing to accept some loss of privacy in return for a faster "anomaly detection" check, however information of people used to develop the data mining model (who have nothing to gain from the faster check) must not be disclosed. Such asymmetric privacy and security requirements raise new challenges.

6. Mining Data Streams

In recent years, advances in hardware technology have made it easy to store and record numerous transactions and activities in everyday life in an automated way. Such processes result in data streams that often grow without limit, referred to as data streams. Stream data could come from sensors, video, and other continuous media including transactions. Some research has been performed on mining stream data. Several important problems recently have been explored in the data stream domain. Clustering, projected clustering, classification, and frequent pattern mining on data streams are a few examples.

Clustering is a form of data management which must be undertaken with considerable care and attention. The idea behind clustering is that a given set of data points can be organized into groups of similar objects through the use of a distance function. By defining similarity through a distance function, an entire data stream can be partitioned into groups of similar objects. Methods that do this view the problem of partitioning the data stream into object groups as an application of a one-pass clustering algorithm. This has some merit, but a more careful definition of the problem, with far better results, will view the data stream as an infinite process with data continually evolving over time. Consequently, a process is needed which can, *de novo* and continuously, establish dominant clusters apart from distortions introduced by the previous history of the stream. One way to accomplish this is to resize the data set periodically to include new data sampling and processing from time to time. The operator could set parameters such as how to include old data processed along with the new, at what level of granularity and during what time period. It is important that past discoveries do not bias future searches that could miss newly formed (or rarely formed) clusters.

An interesting proposal for a two component process for clustering data streams is found in Aggarwal et al. [AHWY03], where the components are an on-line micro-clustering process and an off-line macro-clustering process. The first of these, the on-line micro-clustering component, is based on a procedure for storing appropriate summary statistics in a fast data stream. This must be very efficient. The summary statistics consist of the sum and square of data values, and is a temporal extension of the cluster feature vector shown by BIRCH [ZRL96] With respect to the offline component, user input is combined with the summary statistics to afford a rapid understanding of the clusters whenever this is required, and because this only utilizes the summary statistics it is very efficient in practice.

Flexibility to consider the way in which the clusters evolve over time is a feature of this two-phased approach, as is an opportunity for users to develop insight into real applications. The question of how individual clusters are maintained online is discussed in Aggarwal et al. Using an iterative approach, the algorithm for high-dimensional clustering is able to determine continuously new cluster structures, while at the same time re-defining the set of dimensions included in each cluster. A normalization process, with the aim of equalizing the standard deviation along each dimension, is used for the meaningful comparison of dimensions. As the data stream evolves over time, the values might be expected to change as well, making it necessary to re-compute the clusters and the normalization factor on a periodic basis. A period for this re-computation can be taken as an interval of a certain number of points.

For projected clustering on data streams, Aggarwal [AHWY04b] et al. have propose a method of high-dimensional projected-data-stream clustering called “HPStream.” HPStream relies upon an exploration of a linear update philosophy in projected clustering, achieving both high scalability and high clustering quality. Through HPStream, consistently high clustering quality can be achieved due to the program’s adaptability to the nature of the real data set, where data their reveal its tight clustering behavior only in different subsets of dimension combinations.

For classification of data streams, Aggarwal [AHWY04a] et al. propose data-stream mining in the context of classification based on one-pass mining. Changes that have occurred in the model since the beginning of the stream construction process are not generally recognized in one pass mining. However, the authors propose the exploitation of incremental updating of the classification model, which will not be greater than the best sliding window model on a data stream, thus creating micro-clusters for each class in the training stream. Such micro-clusters represent summary statistics of a set of data points from the training data belonging to the same class, similar to the clustering model in the off-line component [AHWY2003]. To classify the test stream in each instance, a nearest neighbor classification process is applied after identifying various time horizons and/or segments. When different time horizons determine different class labels, majority voting is applied.

With regard to frequent pattern mining on data stream, Han and his colleagues (see [GIAN04]) discuss algorithms at multiple time granularities. They first discuss the landmark model of Motwani and others [DATA03] and argue that the landmark model considers a stream from start to finish. As a result the model is not appropriate for time sensitive data where the patterns such as video patterns as well as transactions may be sensitive to time. Therefore, they focus on data streams over certain intervals depending on the time sensitivity and describe algorithms for extracting frequent patterns in stream data. In particular they consider three types of patterns; frequent patterns, sub frequent patterns, and infrequent patterns. They argue that due to limited storage space with sensor devices, one cannot handle all kinds of patterns. Therefore, they focus on frequent patterns and sub frequent patterns as the sub frequent patterns could become frequent patterns over time. They illustrate tree-building algorithms, which essentially develop a structure that is a pattern tree with a time window. Such a structure is what they call an FP-Stream. This technique essentially relies on the FP-streams.

Besides these, Demers, Gehrke et al. use the notion of an information sphere that exist within an agency and focused on mining the multiple high-speed data streams within that agency [DGR04]. They also discuss the global information spheres that span across the agencies and focus on joining multiple data streams.

One major difference is noted between what we have called real-time data mining and the data-stream mining defined by Han and others. In the case of real-time data mining, the goal is to mine the data and output results in real-time. That is, the data-mining algorithm must meet timing constraints and observe deadlines. In the case of stream mining, the goal is to find patterns over specified time intervals. That is, the patterns may be time sensitive but the result may not necessarily lead to an urgent action on the part of a decision maker unless the pattern were to emerge in time to allow appropriate follow-up action. We can also see the similarities between the two notions. That is, while stream mining has to find patterns within a specified time interval, it may also imply that after the interval has passed, the patterns may not be of much value. That is, stream mining also has to meet timing constraints in addition to finding patterns with time sensitive data. Essentially what we need is a taxonomy for real-time data mining that also includes stream mining.

7. Summary and Directions

We introduced the notion of real-time data mining in [TCCM01]. Whereas the focus of that paper was on mining multimedia data, in this paper we discuss dependability issues for data mining. Recently much emphasis has been placed on data mining algorithms meeting timing constraints as well as mining time-sensitive data. For example, how much do we lose by imposing constraints on the data mining algorithms? In some situations it is critical that analysis be completed and the results reported within few seconds rather than, say, a few hours.

We first discussed issues of real-time data mining and then we examine various data mining techniques such as associations and clustering and discussed how they may meet timing constraints. We also discussed issues of using parallel processing techniques and mining data streams. This is because streams come from sensor and video devices and the patterns hidden in the streams may be time sensitive. We also discussed dependability issues for data mining.

Since we introduced the notion of real-time data mining in [TCCM01] much interest has emerged in the field. Many applications, including counter-terrorism and financial analysis, clearly need this type of data mining.. Our paper has provided some initial directions. M any opportunities and challenges remain in real-time data mining.

Acknowledgements

The authors thank Dr. Rick Steinheiser for valuable comments. The authors also thank the Office of Naval Research and the SSC-SD Science and Technology initiative for financial support. This work is approved for public release with an unlimited distribution. Finally the authors thank Lei Wang for formatting the paper.

References

[AHWY03] C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A Framework for Clustering Evolving Data Streams", Proc. 2003 Int. Conf. on Very Large Data Bases (VLDB'03), Berlin, Germany, Sept. 2003.

[AHWY04a] C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "On Demand Classification of Data Streams", Proc. 2004 Int. Conf. on Knowledge Discovery and Data Mining (KDD'04), Seattle, WA, Aug. 2004.

[AHWY04b] C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A Framework for Projected Clustering of High Dimensional Data Streams", Proc. 2004 Int. Conf. on Very Large Data Bases (VLDB'04), Toronto, Canada, Aug. 2004.

[AS00] Rakesh Agrawal and Ramakrishnan Srikant, "Privacy-Preserving Data Mining", Proceedings of the 2000 ACM SIGMOD international conference on Management of data, Dallas, Texas, pp.439 – 450.

[AX99] S. Axelsson, Research in Intrusion Detection Systems: A Survey,. TR 98-17 (revised in 1999), Chalmers University of Technology, 1999.

[CHNW96] D.W. Cheung, J. Han, V. Ng, and C.Y. Wong, Maintenance of discovered association rules in large databases: An incremental updating technique. In Proc. 1996 Int. Conf. Data Engineering, pages 106 -114, New Orleans, Louisiana, Feb. 1996.

[CL03] Chris Clifton, "Change Detection in Overhead Imagery using Neural Networks", International Journal of Applied Intelligence 18(2), Kluwer Academic Publishers, Dordrecht, The Netherlands, March 2003.

[CWYM04] Yun Chi Haixun Wang, Philip S. Yu, Richard R. Muntz, Moment: Maintaining Closed Frequent Itemsets over a Stream SlidingWindow , ICDM'04.

[DGR04] Demers, A., J. Gehrke, and M. Riedewald, Research Issues in Mining and Monitoring Intelligene Data, Next Generation Data Mining, AAAI Press, 2004. (Editors: H. Kargupta et al)

[LF01] W. Lee and W. Fan, Mining System Audit Data: Opportunities and Challenges, SIGMOD Record 30(4):33-44, 2001.

[DATA03] M. Datar, A. Gionis, P. Indyk, and R. Motwani, Maintaining stream statistics over sliding windows. Proc. 13th SIAM-ACM Symp. on Discrete Algorithms, 2002.

[TCCM01] Thuraisingham, B., C. Clifton, M. Ceruti, and J. Maurer, Real-time Multimedia Data Mining, Proceedings of the ISORC Conference, Magdeberg, Germany, 2001.

[THUR03] Thuraisingham, B., Data Mining for Business Intelligent and Counter-terrorism, CRC Press, 2003.

[VC04] Jaideep Vaidya and Chris Clifton, "Privacy-Preserving Data Mining: Why, How, and What For?", IEEE Security & Privacy, New York, NY, November/December, 2004.

[WS04] K. Wang & S. Stolfo, Anomalous Payload-based Network Intrusion Detection, RAID, 2004.

[ZRL96] T. Zhang, R. Ramakrishanan, M. Livny, "BIRCH: An Efficient Data Clustering Method for Very Large Databases," Proc. of ACM SIGMOD Conference, 1996.