MAIDS: Mining Alarming Incidents from Data Streams*

Y. Dora Cai[§] David Clutter[§] Greg Pape[§] Jiawei Han[†] Michael Welge[§] Loretta Auvil[§]

[§] Automated Learning Group, NCSA, University of Illinois at Urbana-Champaign, U.S.A.
[†] Department of Computer Science, University of Illinois at Urbana-Champaign, U.S.A.

1. INTRODUCTION

Many applications exist today that require the analysis of data streams. Data streams are dynamically changing, in high volume, potentially infinite, and require *multi-dimensional analysis*. These unique characteristics have posed great challenges for data analysis in this area.

This paper presents a demonstration of our recent research and development of the MAIDS system, which mines alarming incidents from data streams, with the following major analysis functions: (1) multi-resolution modeling using a tilted time window framework, (2) multi-dimensional analysis using a stream "data cube" model, (3) online stream classification, (4) online frequent pattern mining, (5) online clustering of data streams, and (6) stream mining visualization.

2. SYSTEM ARCHITECTURE

The architecture of MAIDS is shown in Figure 1. The top box shows incoming data streams from various applications that produce data streams indefinitely. After data preprocessing, such as data formatting, normalization, and transformation, the data streams are simultaneously sent to the following four modules: *Stream Query Engine*, *Stream Data Classifier*, *Stream Pattern Finder*, and *Stream Cluster Analyzer*, which will generate query results, classification models, frequent patterns, and data clusters, respectively. These results are output to users in various forms including graphs and charts generated by the *Stream Mining Visualizer*.

There are several unique features of the MAIDS system. First, the system adopts a flexible *tilted time window* framework throughout all of the functional modules. This framework [2, 3] was developed based on the assumption that in stream data analysis, recent data is usually more important than historical data. Thus, the most recent time can be registered at the finest granularity, and the more distant the coarser granularity. The level of coarseness depends on application requirements.

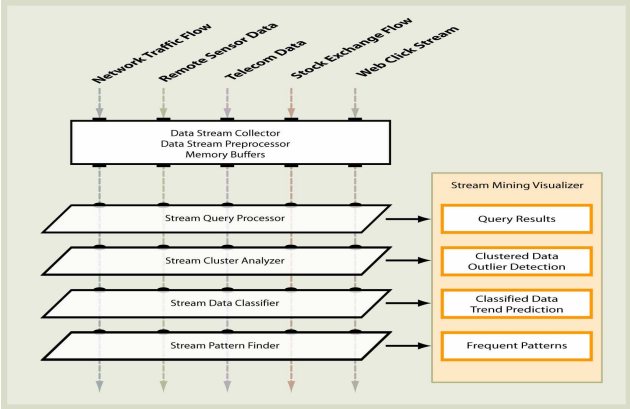

Figure 1: MAIDS Architecture

The current implementation of MAIDS adopts the *natural tilted time window*, that uses natural time to configure the time granularity (e.g., second, minute, hour, etc.). However, since it is implemented in the object-oriented programming methodology, this model can be changed easily into other models, such as *logarithmic tilted time window*, and *pyramidal tilted time window* based on the application requirements [1]. The *natural tilted time window* is implemented using circular queues. Each queue is for a specific time granularity. The *tilted time window* is automatically self-maintained. Whenever reaching the boundary of a time granularity, the aggregates stored in a finer granularity level are summarized and propagated to a coarser granularity level. This technique has substantially compressed the data without losing important information, and thus has made possible long-running analyses on data streams.

Second, the system facilitates multi-dimensional analyses using a stream cube architecture [3]. This architecture ensures that online analytical processing can be performed on stream data in a similar way as OLAP in data cubes.

Third, the system integrates multiple stream mining functions into one platform so that multiple mining functions can cooperate to discover patterns and alarming incidents in real time.

* The work was supported in part by U.S. Office of Naval Research and National Science Foundation(IIS-03-08215).

SIGMOD 2004 June 13-18, 2004, Paris, France.

MAIDS is a general-purpose tool for data stream analysis and is designed to process high-rate and multi-dimensional stream data. MAIDS can interface with data streams generated by various types of devices and has many applications, such as network intrusion detection, telecommunication data flow analysis, credit card fraud prevention, Web clickstream analysis, and financial data trend prediction. MAIDS has been integrated into the D2K (Data to Knowledge) framework [9]. D2K is being developed by Automated Learning Group, NCSA, at the University of Illinois at Urbana-Champaign.

3. MAJOR FUNCTIONAL COMPONENTS

MAIDS consists of five functional modules. The *Stream Query Engine* serves as a powerful stream query processor that supports many query options, including *single-dimensional* vs. *multi-dimensional* queries, *ad-hoc* vs. *continuous* queries, *drill-down* vs. *roll-up* OLAP queries, *point* vs. *duration time* queries, and *exact* vs. *approximate* queries. The query result can be either presented in a report format or visualized in a chart or graph. The *Stream Query Engine* has several novel features. It dynamically constructs a largely virtual *stream data cube* using the H-tree data structure [5, 3]. It only materializes the internal nodes of the tree along the popular querying path between two layers: (i) *minimum-interest layer*, and (ii) *observation layer*. Queries that fall outside of the popular querying path are answered by minimal computation on target cells from those reachable at run-time. Each tree node in the H-Tree stores a *tilted time window* that tracks the aggregates (count, sum, max, min) in different time slots for the node.

The *Stream Data Classifier* constructs classification models dynamically based on the current as well as historical stream data collected in the *tilted time window*. We have integrated the Naïve Bayesian algorithm [8] with modifications made for stream data analysis. Several distinct features can be identified in our design and implementation. An efficient data structure, called *AVC-list* [4], has been constructed dynamically and maintained incrementally to accumulate single variable statistics for the Naïve Bayesian classifier, and a *tilted time window* is associated with each node in the AVC-list. The classification models can be built automatically at the requested time horizons and at the specified time intervals. Many techniques have been applied during model building to promote model accuracy, such as multi-model evaluation and boosting. The model constructed can be immediately applied to predict events for incoming data streams.

The *Stream Pattern Finder* has been constructed to discover frequent patterns. Data streams may contain many hidden patterns. The underlying algorithm essentially adopts the extended frequent pattern growth approach [6] which discovers frequent patterns for the interested sets of items specified by users. This module dynamically constructs an FP-tree while scanning data streams. Each node in the tree contains a *tilted time window* that accumulates counts of the frequent patterns for each time slot. The frequent patterns and association rules for a requested time horizon can be extracted and visualized using the FP-tree structure.

The *Stream Cluster Analyzer* dynamically performs cluster analysis based on the current data set as well as those stored in different slots in the *tilted time window*. This method is based on our research and essentially constructs clusters in two steps: (1) micro-clustering, and (2) macro-clustering. The first step dynamically builds a hierarchical CF-Tree similar to BIRCH [10]. However, each entry of a leaf node in the CF-Tree stores a *tilted time window*, each time slot of the *tilted time window* holds a *CF-Vector*, and the entries in the leaf nodes form micro-clusters. The second step uses a modified k-means algorithm [7] to compute the macro-clusters for a requested time horizon. The computation applies several techniques, such as seed sampling, distance-based partition, and weighted centroid adjustment.

The *Stream Mining Visualizer* presents visualizations of the other modelling tools. The visualizations are updated on demand or continuously, depending on which mining tool it is associated with. They are the watchdogs of the dynamic streaming system and will trigger alarms and give messages when alarming incidents are detected from the ongoing stream data.

4. CONCLUSIONS

The MAIDS system (http://maids.ncsa.uiuc.edu) is a joint R & D effort between the Automated Learning Group, NCSA and the Department of Computer Science at the University of Illinois at Urbana-Champaign. It is a special component of D2K for stream data analysis. The MAIDS system has shown excellent accuracy and performance on real applications. Using the network flow analysis as an example, we have been able to detect most network intrusions by applying these analysis tools.

5. REFERENCES

[1] C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for clustering evolving data streams. VLDB'03.

[2] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. PODS'02.

[3] Y. Chen, G. Dong, J. Han, B. W. Wah, and J. Wang. Multi-dimensional regression analysis of time-series data streams. VLDB'02.

[4] S. Guha, R. Rastogi, and K. Shim. Cure: An efficient clustering algorithm for large databases. SIGMOD'98.

[5] J. Han, J. Pei, G. Dong, and K. Wang. Efficient computation of iceberg cubes with complex measures. SIGMOD'01.

[6] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. SIGMOD'00.

[7] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer-Verlag, 2001.

[8] T. M. Mitchell. *Machine Learning.* McGraw Hill, 1997.

[9] M. Welge, et al. Data to knowledge (D2K), A rapid application development environment for knowledge discovery in databases. *NCSA Technical Report*, 2003.

[10] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: an efficient data clustering method for very large databases. SIGMOD'96.