Exploring Multivariate Data Streams Using Windowing and Sampling Strategies

Zaixian Xie
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609 USA
xiezx@cs.wpi.edu

Matthew Ward
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609 USA
matt@cs.wpi.edu

Elke Rundensteiner
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609 USA
rundenst@cs.wpi.edu

CHI 2008, April 5 – April 10, 2008, Florence, Italy
ACM 978-1-60558-012-8/08/04.

Abstract

The analysis of data streams has become quite important in recent years, and is being studied intensively in fields such as database management and data mining. However, to date few researchers in data and information visualization have investigated the visual analytics of streaming data. Although streaming data is similar to time-series data, its large-scale and unbounded characteristics make regular temporal data visualization techniques not effective.

In this paper, we propose a framework to visualize multivariate data streams via a combination of windowing and sampling strategies. In order to help users observe how data patterns change over time, we display not only the current sliding window but also abstractions of past data in which users are interested. Uniform sampling is applied within a single sliding window to help reduce visual clutter as well as preserve the data patterns. However, we allow different windows to have different sampling ratios to reflect how interested the user is in the contents. To achieve this functionality, we propose to use a DOI (degree of interest) function to represent users' interest for the data in a particular sliding window. In order to visually

convey the multi-correlations and trends at the same time, we use multiple views, the union of traditional multivariate visualizations and time charts.

Keywords
Data stream, multivariate data, sampling, windowing

ACM Classification Keywords
H5.2. Information Interfaces and Presentation: User Interfaces—– Graphical user interfaces

Introduction

Advances in hardware enable people to record data at rapid rates, e.g. kilobytes per second or even higher speeds. Many real world applications require data collection at such a high speed. Moreover, the newly acquired or generated data items often need to be processed immediately. In the areas of database and data mining, the term *data streams* has been introduced to refer to such data that keeps growing and needs to be processed on the fly. Researchers have developed a lot of techniques to manage, query and analyze data streams and make decisions in real-time.

Since windowing strategies are commonly employed in the management of data streams, a naive solution would be to split the whole stream into contiguous sliding windows and send each through the visualization pipeline one by one. The obvious disadvantage of this technique is that only the current fragment of the data stream is displayed, and it is difficult to learn how data patterns have changed over time because the past data is no longer visible. Another solution is using traditional time-series data visualization techniques. However, the commonly used abstraction techniques for static datasets, such as sampling, must be adapted to unbounded streams. Otherwise, the system will become less efficient if we keep restarting the data abstraction algorithm when new data items are available.

In this paper, we combine windowing and sampling strategies to preprocess multivariate data streams, and then visualize it using multiple views. We display not only the data in the current sliding window, but also abstractions of past data in the same or separate views, helping users learn how data patterns change over time. For each sliding window, we allow different sampling ratios based on the degree of users' interest, which is determined by a DOI function (Degree of Interest) [1,2] defined by users.

We will focus on a special type of data stream, namely univariate-aggregation, in which each data item is a multi-dimensional vector and each dimension can be regarded as univariate data. This type is very common. One example is a stock price data stream in which each dimension represents the price of one company's stock. Another is a set of sensors monitoring the status and positions of military vehicles. The main contributions of this paper include:

- We propose a framework to introduce windowing and sampling strategies into traditional multivariate and time-series data visualization techniques. This combination is aimed at handling unbounded input.
- This framework allows users to define a DOI function to describe the degree of users' interest for different portions of the data.
- Users can choose multiple views to observe the data stream, including traditional multivariate and

time-series visualizations. Linked interactions among these views are provided to help users detect and isolate data patterns and their changes.

The Framework Based on Sampling and DOI Functions

Figure 1 shows the proposed framework. Here we use non-overlapped sliding windows. Because a data stream is infinite in nature, the data is sampled first and then saved in the "data item memory pool". When the memory pool is full, a part of the old data will be moved to the disk pool. The "mixer" is the core of the whole framework. Its input includes the current window and segments of old windows. It can assign sampling ratios to these windows and generate outputs that mix datapoints from these windows. Users can observe data from multiple views, such as scatterplot matrices for the representation of multi-correlations, and time charts to convey the trend of each dimension.

To determine the sampling ratios for current and old windows in the mixer, we introduce a DOI (Degree of Interest) function to represent how interested the user is in seeing a particular sliding window. The DOI function has two parameters, a timestamp t_d for a specific sliding window, and the current time point t_c. Formally, the DOI function is given as: $DOI = f_{doi}(t_d, t_c)$. The DOI value will be mapped to a sampling ratio based on the rule that the portions in which users have higher interest will be shown in more detail.

In order to help users define DOI functions, we provide an interactive chart. A curve corresponding to the DOI values for past sliding windows will be displayed in an accompanying window. When the user is monitoring data streams, he can adjust this curve for more or less detail in specified portions.

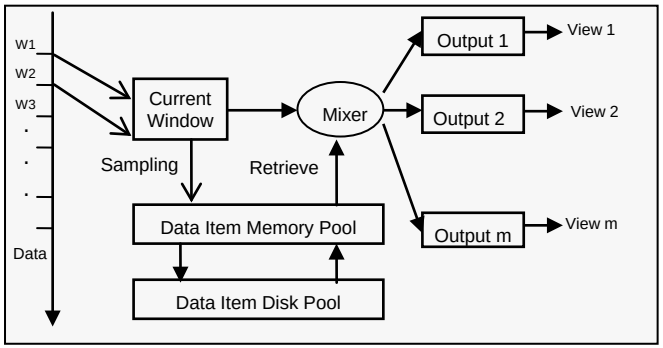

Figure 1. The framework of user-driven multiple-view visualization for data streams.

For convenience, we also propose two types of DOI functions used for normal circumstances. One is used to observe how data pattern changes across several sliding windows, namely **PC** (Pattern Change) Function. The other aims to convey repeated phenomena, which we call **PP** (Periodic Phenomena) function. Figures 2 and 3 show examples of these functions.

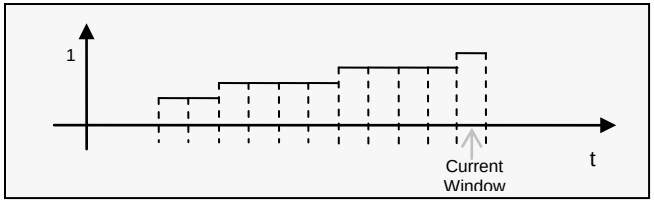

Figure 2. The DOI function for pattern change.

Views

We propose three types of views for multi-resolution data streams after windowing and sampling.

- A general adaptation of traditional multivariate visualizations: We regard the output of the "mixer" as a dataset and display it using traditional multivariate visualization techniques, such as parallel coordinates and scatterplot matrices. Available visual attributes are used to convey the timestamps of datapoints. For example, we can use dot size, color or opacity to show the age of data items in scatterplot matrices.

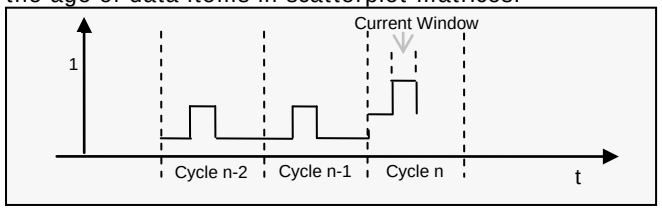

Figure 3. The DOI function for periodical phenomena.

- Scatterplot Matrices with Diagonal Plots: We show line charts for each dimension and put them in the diagonal plots of a scatterplot matrix. This makes it easy to convey the trends for each dimension without the need for more canvas space (See Figure 4).
- Multiple Views: A couple of views can be generated at the same time. Some views are used to convey trends, while others could represent multi-correlations.

We make heavy use of linked brushing in our framework. Users can select a subset of data in one view, e.g., through a range query, and the system will highlight this subset in all views. As we know, different techniques can show different data patterns. For example, time charts excel in representing trends, and scatteplot matrices make it easy to observe correlations between any two dimensions. Thus this interaction can help users retrieve data patterns in data streams.

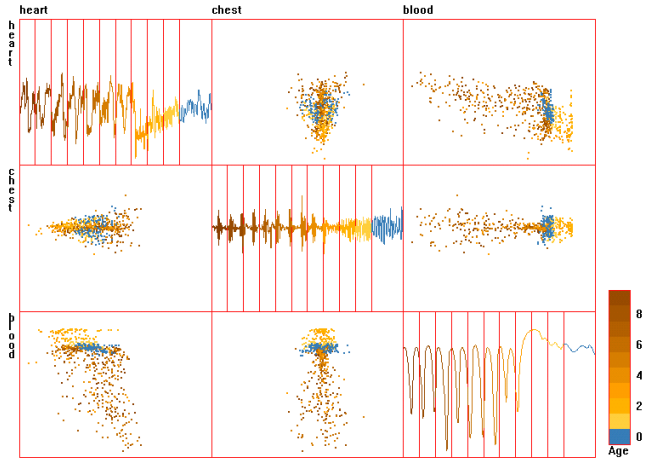

Figure 4 The dot color denotes the age of the datapoint (the difference between its timestamp and current time).

References

[1] Furnas, G.: Generalized fisheye views. Proc. ACM SIGCHI Conference on Human Factors in Computing Systems (1986), 16–23.

[2] Hao, M. C., Dayal, U., Keim, D. A., and Schreck, T. Multi-resolution techniques for visual exploration of large time-series data. EuroVis07: Joint Eurographics – IEEE VGTC Symp. on Visualization, pages 27–34, 2007.