A System for Mining Correlations among Multiple Evolving Data Streams

Wei Fan [*1] Toyohide Watanabe [*1] Koichi Asakura [*2]

[*1]Graduate School of Information Science, Nagoya University
[*2]School of Informatics, Daido University

We propose a framework to monitor multiple evolving numeric data streams, and determine which pairs are correlated with lags, as well as the values of lags. Traditional algorithms resolved the lag correlation mining of whole data streams. While taking into account of the existence of concept drifts in evolving data streams, it is interesting to discover the correlation among subsequences in the continuous data. In this paper, we detect the drifting concepts in data and perform lag correlation among subsequences without any concept drifts in order to generate more accurate results. Our framework can handle data streams of semi-infinite length, incrementally, efficiently, and with small resource consumption, based on a technique to summarize a dynamic data stream incrementally at multiple resolutions. According to Nyquist's sampling theorem, the framework can estimate lag correlations based on the summarization statistics of data streams with little, and often no error at all.

1. Introduction

In this paper, we focus on *lag correlations* mining among multiple evolving data streams. We propose a framework to determine automatically all the pairs of data streams that correlated with time-delaies (lags), as well as to report the values of lags. Lag correlations are frequent in practices: in a sensor network, the measurements at two nodes have a lag corrlation of l, that is, the two sequences look very similar when one is delayed by l time-ticks, due to different speed for transforming sensor data from nodes to process center. The knowledge of lag corrrelation can be used for prediction and is helpful to discover potential anomalies.

Traditional algorithms resolved the problem of lag correlation mining among whole data streams, ignoring the changes happen in the continuous data. While in most cases, a change happens in one of the sequences, which triggers a change in the trend of another sequence at the same time or after a time delay. For example, a decrease in interest rate typically precedes an increase in house sales by a few months; higher amounts of fluoride in the drinking water may lead to fewer dental cavities, some yeas later. Therefore, we detect the change in data automatically and perform lag correlation among subsequences after changes happen in order to generate more accurate results. Our framework also address the critical time and space constraints in the data stream environment. Characteristics of our framework can be summarized as follows:

- **Change detection.** In our framework, we monitor the correlation coefficent respect to lag l which maxmizes the correlation coefficient in the previous data. Intuitively, a change happens if the correlation coefficient respect to lag l decreases. Compared with traditional algorithms treating whole data streams or sliding windows, our framework can perform lag correlation among data streams with different evolving speeds and is able to generate more accurate resutls of correlation analysis.

- **Incremental computation.** The computation time of the lag correlation coefficient in terms of a specified lag l per time tick is constant, which satisfies the time constraint in data stream environment.

- **Estimation of correlation coefficients.** In order to find the maximum correlation coefficient, we need to calculate the correlation coefficients respceting to lag l which grows geometrically up to the length of sequences. In the case of subsequences continuously increase in length (e.g. stable data streams), It is challenging. In this paper, we prob the lag correlation coefficients at values of the lag l that form a geometric progression. Thus, we need only $O(\log n)$ numers to estimate the lag correlation coefficient.

- **Accuracy.** According to the theoretical analysis in [Sakurai 05], based on Nyquist's sampling theorem, the estimation of lag correlations with litte, and often no error at all.

- **Space complexity.** Corresponding to the geometric progression of lag l, we propose a technique to summarize a dynamic data stream incrementally at multiple resoutions on which the lag correlation analysis are performed. Here the computational cost of incremental summarization is $O(1)$ for each new data point, and the space complexity for calculating the correlation coefficients is dramatically reduced from $O(n)$ to $O(\log n)$.

2. Related Work

There are several related work focus on correlation among data streams. Zhu et al. [Zhu 02] monitors multiple streams in real time. They use the "short window" Fourier Transform to summarize streams, and then computer all the pairwise correlations and lag correlations. However, correlation

Contact: Wei Fan, Graduate School of Information Science, Nagoya University, Tel: (052)789-2735, Fax: (052)789-3808, E-mail: fan@watanabe.ss.is.nagoya-u.ac.jp

is defined within a sliding window, therefore, the choice of sliding window is nontrivial in order to discover the correlation among the continuous data streams with different evolving speeds. Additionally, the method will clearly miss any lag correlation that is longer than the window w of the short-window Fourier Transform.

Sakurai et al. [Sakurai 05] proposes an incremental and efficient algorithm for lag correlation mining in data streams with small resource consumption. In this paper, according to the theoretical analysis, probing the lag correlation coefficients at values of lag l in a geometric progression produces little and often no error at all. While the correlation analysis is performed on whole data streams, not considers the correlation of subsequences triggered by changes in the trends of data.

In most applications, the data in data streams change over time, so the correlation among data streams may change due to the evolutions in data. Sliding window based analysis is difficult to perform correlation whenever it is necessary. Yeh et al. [Yeh 07] proposes a framework for clustering multiple data streams based on events and correlations. The mechanism for event or change detection is based on approximation of data streams incrementally by linear segments, and the a change point is consider as the start point of a segment with different slope. There are other researches discuss the concept drifts in data stream mining, such as [Wang 03], [Chen 09] and so on.

3. Change Detection

3.1 Lag Correlation Coefficient

A data stream X is a discrete sequence of numbers $x_1, x_2, \ldots, x_n, \ldots$, where x_n is the most recent value. Notice that n increases with every new time-tick. The definition of the correlation coefficient $R(0)$ between two time sequences X and Y of equal length n and zero lag, is a traditional one, known as Pearson's ρ coefficient:

$$\rho = R(0) = \frac{\sum_t ((x_t - \bar{x}) * (y_t - \bar{y}))}{\sigma(x) * \sigma(y)} \tag{1}$$

where $\bar{x}$, $\bar{y}$ denote the mean of X and Y, respectively. For lag $l (l > 0)$, we consider only the common part of X and the shifted Y; that is, only $n-l$ time ticks, and the equation becomes

$$R(l) = \frac{\sum_{t=l+1}^{n} (x_t - \bar{x})(y_t - \bar{y})}{\sqrt{\frac{1}{n-1}\sum_{t=l+1}^{n} (x_t - \bar{x})^2}\sqrt{\frac{1}{n-1}\sum_{t=1}^{n-l} (y_t - \bar{y})^2}} \tag{2}$$

$$\bar{x} = \frac{1}{n-l}\sum_{t=l+1}^{n} x_t, \bar{y} = \frac{1}{n-l}\sum_{t=1}^{n-l} y_t \tag{3}$$

where $R(l)$ denots the correlation coefficient, when X is delayed by l.

3.2 Subsequence Lag Correlation Analysis

Given a simple example of concept drift in sequences as shown in Fig: 1. At time tick $t = 5$, sequence X has a shift change. At each time tick, we monitor the lag correlation

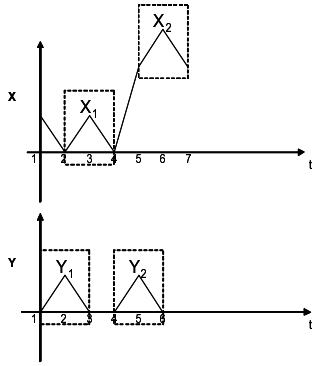

Figure 1: The whole sequences corrlation and subsequences correlation in an example of a concept drift existing in sequences.

coefficient. According to the definition of lag correlation coefficient, we can get the results that at $t = 4$, maximum of $R(l)$ equals 1 at the value of l equals 1, and $t = 7$, maximum of $R(l) = 1$ at the value of l equals 4. The two maximums of correlation coefficient corresponds to the lag correlated pairs of subsequences $\{X_1, Y_1\}$ and $\{X_2, Y_1\}$. But in fact, we may interested in the correlation of pattern X_2 and Y_2 after discovering the lag correlation of pairs $\{X_1, Y_1\}$. Here we can see for the reason that from time tick $t = 5$, the shift pattern change from X_1 to X_2, the lag correlation between X and Y changes (until time tick $t = 4$, Y is correlated with X with time delay equaling 1). In order to resolve this problem, we propose a method to detect the change of pattern automatically. As until $t = 4$, the lag correlation coefficient $R(l)$ respecting to $l = 1$ remains the maximum value, while at $t = 5$, the $R(l = 1)$ decreases. That is to say, at $t = 5$, the lag correlation between X and Y is destroyed, therefore, it is a trigger for a new analysis of lag correlation. Then we calculate $R(l)$ from $t = 5$ of sequence X and $t = 4$ of sequence Y, and reset $l = 1$. Finally, we can discover the lag correlation between subsequence X_2 and Y_2. In conclusion, we monitor the correlation coefficient respect to lag l which maximizes the correlation coefficient in the previous data, and detect a change of pattern if the correlation coefficient respect to lag l decreases.

4. Incremental Computation of Correlation Coefficients

As discussed in [Sakurai 05] and [Yeh 07], the correlation coefficients can be computed incrementally. Next we disscuss the incremental calculation process. Let $S_x(1, n)$ be the sum of sequence X of length n (i.e. $S_x(1, n) = \sum_{t=1}^{n} x_t$), and $S_{xx}(1, n)$ be the sum of the squares of X (i.e. $S_{xx}(1, n) = \sum_{t=1}^{n} x_t^2$). $S_{xy}(l)$ means the inner-product for X and the shifted sequence Y:

$$S_{xy}(l) = \sum_{t=l+1}^{n} x_t y_{t-l} \tag{4}$$

We shall refer to all these values collectively as sufficient statistics. Given our sufficient statistics, the correlation coefficient $R(l)$ is obtained by

$$R(l) = \frac{C(l)}{\sqrt{V_x(l+1,n) * V_y(1,n-l)}} \quad (5)$$

where $C(l)$ is the covariance of X and Y:

$$C(l) = S_{xy}(l) - \frac{S_x(l+1,n) * S_y(1,n-l)}{n-l} \quad (6)$$

and $V_x(l+1,n)$ means the variance of subsequence of X, starting from $t = l+1$:

$$V_x(l+1,n) = S_{xx}(l+1,n) - \frac{(S_x(l+1,n)^2)}{n-l} \quad (7)$$

The variance $V_y(1,n-l)$ of Y is computed similarly. In conclusion, for the given value of lag l, we only need to keep track of five numbers, the sufficient statistics, because they are enough to help us estimate the correlation $R(l)$, at any point of time.

5. Estimation of Correlation Coefficients

Although the calculation of correlation coefficient at each time tick is incremental, we also aim to prob the lag l in order to maximize the lag correlation coefficient $R(l)$. Therefore, in order to find the maximum correlation coefficient, we need to calculate the correlation coefficients respecting to lag l which grows geometrically up to the length of sequences. We compare the subsequences between two change points, but in the case of subsequences continuously increase in length (e.g. subsequences of stable data streams), reporting the lag l with maximum $R(l)$ is a challenging problem. Here, refer to the method in [Sakurai 05], specially, instead of computing $R(l)$ for every possible value of lag l, we propose to keep track of only a geometric progression of the lag value: $l = 0, 1, 2, 4, 8, \ldots$. The justification is that it achieves a dramatic reduction in computation time, since we need only $O(\log n)$ numbers to keep track of, instead of $O(n)$ that the "naive Solution" requires. As discussed in theoretical analysis of [Sakurai 05], based on Nyquist's sampling theorem, the estimation of lag correlations with little, and often no error at all.

6. Space Complexity for Estimation of Correlation Coefficient

Corresponding to the geometric progression of lag l, we propose a technique to summarize a dynamic data stream incrementally at multiple resolutions on which the lag correlation analysis is performed. Here the computational cost of the incremental summarization is $O(1)$ for each new data point, and the space complexity is dramatically reduced from $O(n)$ to $O(\log n)$.

We use a average scheme to compute multi-resolution approximations of a single data stream. These approximations

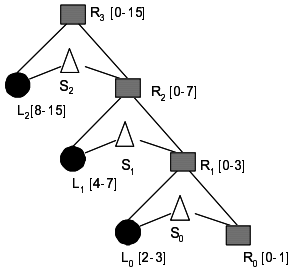

Figure 2: Multi-resolution summaries of a data stream.

are shown pictorially in Fig. 6. for the case $n = 16$. A level 3 approximation denoted as A_3 stores a set of values that summarize $[0, \ldots, 15]$. A level 2 approximate denoted as A_2 stores two sets of values: one summarizing $[0, \ldots, 7]$ and the other summarizing $[8, \ldots, 15]$. There are 3 nodes at each level to keep the corresponding approximations, Left Node (L), Shift Node (S) and Right Node (R) from left to right. The approximation stored at the Right Node will be shifted to the Left Node after some time units to represent an old approximation. Shift Node acts as an intermediary in this process. When a new data point arrives, we update the multi-resolution summaries and choose the Right Nodes to estimate the lag correlation coefficients $R(l), l = 0, 1, 2, \ldots, \log n$. Fig. 3 illustrates an example for update of the multi-resolution approximations at each new data point. At $t = 0$, every node is up-to-date as shown in Fig. 3(a). At $t = 1$, a new data value, 4, arrives. At $t = 1$, L_0 gets the summary stored in S_0, $14/2$, and S_0 gets $26/2$ from R_0. R_0 computes the average of 14 and 4. The average $18/2$ is stored in R_0. All nodes at higher levels are shifted up by 1 time unit. For example, L_2 now stores an approximation to $[9-16]$ instead of $[8-15]$. Fig. 3(b) shows the resulting tree. At $t = 2$, 6 arrives. At level 0, L_0 gets $26/2$ from S_0, and S_0 gets $18/2$ from R_0. The new average of $[0-1]$, $10/2$, is stored in R_0. At level 1, L_1 gets $8/4$ from S_1, and S_1 gets $32/4$ from R_1. Lastly, R_1 computes and stores the average of R_0 and L_0, which is $36/4$. Fig.3(c) shows the resulting tree. Fig. 3(d), Fig. 3(e) and Fig. 3(f) show the resulting tree after the arrival of $2, 10$ and 4. As illustrated, the computation cost for approximating each new arrival data point is $O(1)$. Thus, we use a cubic spline to interpolate the missing correlation coefficients between the approximated coefficients. It effectively estimates that the correlation coefficients vary between these lags. Finally, we can see the lag correlation from the local maximum of the cubic spline curve (solid line in Fig. 7.).

7. Conclusions

In this paper, we extend traditional algorithms for lag correlation mining to treat evolving data streams. It is able to discover the lag correlations among subsequences and report the values of lags automatically. Our framework can handle data streams of semi-infinite length, incrementally, efficiently, and with small resource consumption, based on

a technique to summarize a dynamic data stream incrementally at multiple resolutions. As reference to Nyquist's sampling theorem, the framework can estimate lag correlations based on the summarization statistics of data streams with little, and often no error at all. As the future work, we will apply the proposed framework to synthetic and real data in order to evaluate its effectiveness and efficiency.

References

[Chen 09] Chen. H.L., Chen. M.S., Lin. S.C.: Catching the Trend: A Framework for Clustering Concept-Drifting Categorical Data. IEEE Transactions on Knowledge and Data Engineering, Vol. 21, No. 5, pp. 652–664. (2009)

[Sakurai 05] Sakurai. Y, Papadimitriou. S, Faloutsos, C.: BRAID: Stream Mining through Group Lag Correlations. Proceedings of the ACM SIGMOD international Conference on Management of Data, pp. 599–610. ACM Press, Maryland (2005)

[Wang 03] Wang. H.X., Fan. W., Yu. P.S., Han, J.W.: Mining concept-drifting data streams using ensemble classifiers. Proceedings of the ACM SIGKDD international Conference on Knowledge Discovery and Data mining, pp. 226–235. ACM Press, Washington, D.C. (2003)

[Yeh 07] Yeh. M.Y., Dai. B.R., Chen. M.S.: Clustering over Multiple Evolving Streams by Events and Correlations. IEEE Transactions on Knowledge and Data Engineering, Vol. 19, No. 10, pp. 1349–1362. (2007)

[Zhu 02] Zhu Y.y, Shasha. D: StatStream: Statistical Monitoring of Thousands of Data Streams in Real Time. Proceedings of the VLDB, pp. 358–369. Hong Kong (2002)

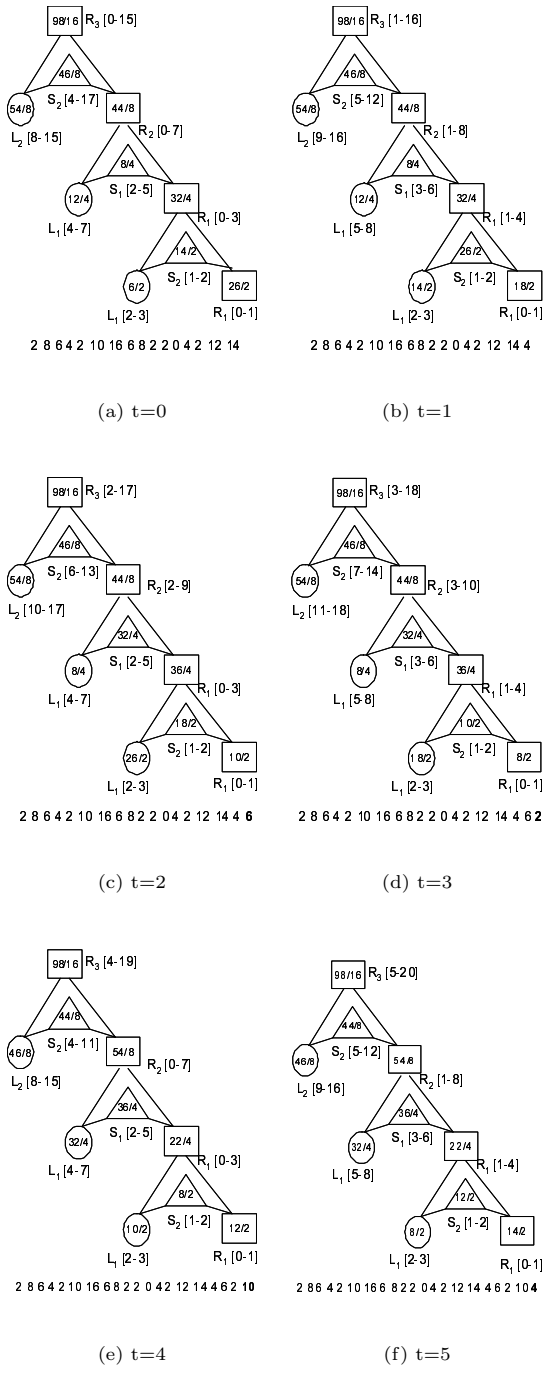

Figure 3: Updata of Multi-resoluion approximations for 5 new data arrivals

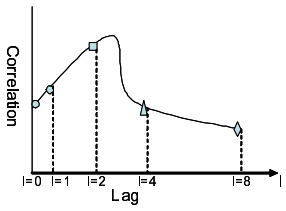

Figure 4: Estimation of correlation coefficients.