World Wide Web (2011) 14:1–25
DOI 10.1007/s11280-010-0095-z

Leadership discovery when data correlatively evolve

Di Wu · Yiping Ke · Jeffrey Xu Yu · Philip S. Yu · Lei Chen

Received: 18 January 2010 / Revised: 17 June 2010 /
Accepted: 21 June 2010 / Published online: 10 July 2010

Abstract Nowadays, World Wide Web is full of rich information, including text data, XML data, multimedia data, time series data, etc. The web is usually represented as a large graph and PageRank is computed to rank the importance of web pages. In this paper, we study the problem of ranking evolving time series and discovering leaders from them by analyzing lead-lag relations. A time series is considered to be one of the leaders if its rise or fall impacts the behavior of many other time series. At each time point, we compute the lagged correlation between each pair of time series and model them in a graph. Then, the leadership rank is computed from the graph, which brings order to time series. Based on the leadership ranking, the leaders of time series are extracted. However, the problem poses great challenges since the dynamic nature of time series results in a highly evolving graph, in which the relationships between time series are modeled. We propose an efficient algorithm which is able to track the lagged correlation and compute the leaders incrementally, while still achieving good accuracy. Our experiments on real weather science data and stock data show that our algorithm is able to compute time series leaders efficiently in a real-time manner

D. Wu · Y. Ke (✉) · J. X. Yu
The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong
e-mail: ypke@se.cuhk.edu.hk

D. Wu
e-mail: dwu@se.cuhk.edu.hk

J. X. Yu
e-mail: yu@se.cuhk.edu.hk

P. S. Yu
University of Illinois at Chicago, Chicago, IL, USA
e-mail: psyu@cs.uic.edu

L. Chen
The Hong Kong University of Science and Technology, Kowloon, Hong Kong
e-mail: leichen@cse.ust.hk

and the detected leaders demonstrate high predictive power on the event of general time series entities, which can enlighten both weather monitoring and financial risk control.

Keywords PageRank · time series · lagged correlation · leadership rank · incremental correlation update

1 Introduction

World Wide Web is full of rich data with various formats, including text data, XML data, multimedia data, time series data, video data, image data, and many others. With the increasing amount of data accumulated on World Wide Web, it is important to analyze the relationships between data objects. In the literature, the relationship between web pages has been well studied by modeling the web as a large graph and computing PageRank to rank the importance of web pages. In this paper, we study the relationship between multiple data streams and propose to discover leaders from them. In empirical research [2, 8, 25], the lagged correlation between two data streams has been well studied and efficient algorithms to discover lagged correlations have also been developed [26]. However, the study on summarizing the relationships across multiple data streams is still lacking. The comprehensive relationships among multiple data streams are very helpful in many applications to monitor and control the overall movement of the entity where the data streams are generated. Three application examples are given as follows.

Earth science: In weather teleconnection network, each stream represents the weather observations (e.g., temperature, pressure and precipitation) [29] of a specific point on the latitude-longitude spherical grids. The lagged correlation between two streams indicates that the weather change in one location can affect the weather in another location with some time delay. By analyzing lead-lag on observations in multiple locations, the earth scientists can understand better from which location the weather phenomena originates and how it evolves.

Finance: The stock market can be modeled as a financial network, in which each stream represents the price of a stock. The lead-lag effect between two streams implies that the price change of one stock influences that of another [25]. In finance crisis, when the market goes down dramatically and the government plans to launch finance bailout, the regulators desire to know the subset of stocks which poses risks (influences) on others and triggers the movement of the whole market. They can then apply a program to these market leaders and control the overall systemic risk.

Blogspace: The world wide web links web blogs together, through which people share information and form evolutionary topics [15, 19]. It is particularly interesting to investigate how a topic emerges [11] and how the discussion propagates through the web links.

In this paper, we study the problem of discovering leaders among a set of time series by analyzing lead-lag relations. We target to extract leaders from multiple time series in a real-time manner. Here, we demonstrate the significance of the problem

and the usefulness of the discovered leaders on a real weather dataset. We analyze the streams of the sea surface temperature (SST) on the Pacific ocean (30° S–30° N, 55° E–80° W) where the famous Nino phenomena occurs irregularly every 4–5 years. We study a period from 1996–1999. In Figure 1, the bold blue line shows the weekly NINO 1+2 index which is a standard weather index developed by earth scientists to study SST anomalies in a Nino region off the coast of Peru. A positive value of the index indicates significant anomalies. As shown in the figure, the NINO 1+2 index begins to increase in January 1997 and goes above 0 in March 1997. Later, it begins to drop and eventually falls below 0 in November 1998. On the other hand, we sample 125 streams of SST time series from that region and extract weekly leaders from them. We then form a leadership index using the extracted leaders weighted by their normalized leadership scores. The red line in Figure 1 gives the leadership index which exhibits a similar but earlier trend to NINO 1+2 index. It begins to increase in September 1996 and rises above 0 in October 1996, which is 20 weeks earlier than NINO 1+2 index. Later, it falls below 0 in July 1998, which is 19 weeks earlier. To further confirm the relationship between the two indices, we conduct a Granger-causality analysis [12] by performing F-test on the lagged value of both indices. After selecting the optimal lagged value for the regression model (lag = 2 for NINO 1+2 index and 1 for leadership index), the result suggests that the leadership index Granger-causes NINO index (the F-Statistics is 6.64) while NINO index does not Granger-cause leadership index (the F-Statistics is statistically insignificant).

Through this example and many other experimental results, we find that the discovered leaders are able to bring enlightening information. First, leaders are good representatives of the whole entity. An event usually introduces some changes to leaders, whose effect then propagates to related time series. As a result, analysts only need to monitor and analyze leaders in order to evaluate the overall entity movement triggered by events. Second, since the leadership is defined by the lagged correlation, leaders have the predictive power within the computed lag as shown in Figure 1. Therefore, analyzing leaders can detect the trend of an event at an early stage. In weather observation and control, this predictive power is very helpful in giving the scientists an early alert on the weather phenomena and allowing them to do better preventions for the coming disasters.

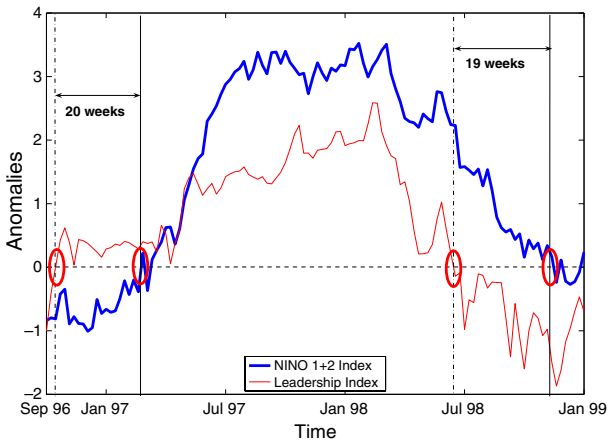


Figure 1 Leadership index vS. NINO 1+2 index from 1996–1999.

The problem of finding the leaders among multiple time series poses great challenges. First, the observations of time series (e.g., temperature, intra-day stock price) usually change rapidly over time, which implies that the leaderships among them may also change from time to time. Therefore, the lagged correlations between pairs of time series, which are used for leadership identification, must be re-computed for every new time tick, while the correlation computation at each time tick is already costly. This high computational complexity makes the design of an efficient solution difficult. Second, after computing the lagged correlation between each pair of streams, how to define and extract useful leaders out of the whole set of time series is also a big challenge.

In this paper, we propose an efficient streaming algorithm to address the problem. The main contributions of the paper are summarized as follows. First, we formalize a new problem of discovering the leadership among multiple time series, which well captures the overall co-movements of time series. Second, we devise an efficient solution that discovers the leaders in a real-time manner. Our solution utilizes an effective update strategy, which significantly reduces the computational complexity in a stream environment. Third, we justify the efficiency of our solution, the effectiveness of our update strategy, as well as the usefulness of the discovered leaders by conducting extensive experiments over the real weather data and financial data.

The rest of the paper is organized as follows. Section 2 gives the preliminaries. Section 3 defines the problem of leadership discovery and discusses the main idea of our solution. Section 4 presents the incremental correlation update strategy. Section 5 reports the performance evaluation. Finally, Section 6 reviews some related work and Section 7 concludes the paper.

2 Preliminaries

We consider a set of N synchronized time series streams $\{S^1, S^2, \ldots, S^N\}$, where each time series $S^j = (s_1^j, \ldots, s_t^j)$ is a sequence of discrete observations over time, and s_t^j represents the value of time series S^j at the most recent time point t. Given a sliding window of length w and a time point t, the sliding window for time series S^j, denoted as $s_{t,w}^j$, is the subsequence $(s_{t-w+1}^j, \ldots, s_t^j)$. Below, we discuss lagged correlation between two sliding windows of two time series streams and the transition probability in a Markov chain which play an important role in this work.

Lagged correlation: The lagged correlation between two sliding windows $s_{t,w}^i$ and $s_{t,w}^j$ of two time series S^i and S^j at lag l, denoted as $\rho_{t,w}^{ij}(l)$, is computed by considering the common parts of the shifted sequences.

$$\rho_{t,w}^{ij}(l) = \begin{cases} \dfrac{\sum_{\tau=t-w+1}^{t-l} \left(s_{\tau+l}^i - \bar{s}_{t,w-l}^i\right)\left(s_\tau^j - \bar{s}_{t-l,w-l}^j\right)}{\sigma_{t,w-l}^i \sigma_{t-l,w-l}^j}, & l \geq 0; \\ \rho_{t,w}^{ji}(-l), & l < 0, \end{cases} \tag{1}$$

where $\bar{s}_{t,w-l}^i$ and $\bar{s}_{t-l,w-l}^j$ are the mean values in the shifted sliding windows $s_{t,w-l}^i$ and $s_{t-l,w-l}^j$, respectively, and $\sigma_{t,w-l}^i$ and $\sigma_{t-l,w-l}^j$ are the standard deviations in $s_{t,w-l}^i$

and $s^j_{t-l,w-l}$, respectively. Here, $\rho^{ij}_{t,w}(0)$ is the correlation with zero lag (known as the local Pearson's correlation [22]). When $l > 0$, $\rho^{ij}_{t,w}(l)$ denotes the correlation between the sliding windows, $s^i_{t,w}$ and $s^j_{t,w}$, when S^i is delayed by a lag l. Therefore, $\rho^{ij}_{t,w}(l)$ is essentially the correlation between the common parts of the shifted windows, $s^i_{t,w-l}$ and $s^j_{t-l,w-l}$, with zero lag. The case when $l < 0$ can be easily handled symmetrically. Since $\rho^{ij}_{t,w}(l)$ is computed on the common parts of the two windows, l is less than the window length w, and in practice $|l| \leq w/2$ as suggested in Box et al. [3].

In a stream context, it is not desirable to compute $\rho^{ij}_{t,w}(l)$ from scratch at each time point t. As shown in Sakurai et al. [26], Zhu and Shasha [33], the lagged correlation can be computed efficiently by tracking some statistics as follows.

$$\rho^{ij}_{t,w}(l) = \frac{\psi^{ij}_{t,w}(l) - \dfrac{\sum\left(s^i_{t,w-l}\right) \cdot \sum\left(s^j_{t-l,w-l}\right)}{w-l}}{\sigma^i_{t,w-l}\sigma^j_{t-l,w-l}}, \tag{2}$$

where $\psi^{ij}_{t,w}(l)$ is the inner product between the shifted windows $s^i_{t,w-l}$ and $s^j_{t-l,w-l}$, $\sum(s^i_{t,w-l})$ and $\sum(s^j_{t-l,w-l})$ are the sum over the two shifted windows, respectively, and $\sigma^i_{t,w-l}$ can be computed as follows.

$$\sigma^i_{t,w-l} = \sqrt{\sum\left(s^i_{t,w-l}\right)^2 - \frac{\left(\sum\left(s^i_{t,w-l}\right)\right)^2}{w-l}}, \tag{3}$$

where $\sum(s^i_{t,w-l})^2$ denotes the sum of the squares of the shifted window $s^i_{t,w-l}$. The value of $\sigma^j_{t-l,w-l}$ can be computed similarly. It implies that as long as the inner product, the sum of squares and the sum of the shifted windows are kept track of, the correlation value at each lag can be computed quickly.

Matrix analysis: In a Markov chain, the transition probability matrix $H = \{h_{ij}\}$ describes the state transition property, where h_{ij} is the probability of being in state j at the next step given that the chain is in state i at the current step. The stationary probability distribution vector for a Markov chain with H is a probability vector $\tilde{\pi}$ such that $\tilde{\pi}H = \tilde{\pi}$. The Markov chain defined by H has a unique stationary probability distribution if H is aperiodic and irreducible (primitive) [1]. Besides, the k-th step probability distribution vector for a chain with N states is defined to be $\pi^k = \{\pi^k(1), \ldots, \pi^k(N)\}$, where $\pi^k(j)$ is the probability of being in state j at the k-th step. Given π^k, the classical power method can be applied to compute π^{k+1} as follows:

$$\pi^{k+1} = \pi^k H. \tag{4}$$

For any starting vector, as long as the transition matrix H is stochastic and primitive, the power method applied to H converges to a unique stationary probability distribution vector $\tilde{\pi}$.

However, the stochastic and primitivity properties are often violated in large graphs, for example, WWW [23, 24]. Therefore, to measure the importance of Web

pages in a large WWW graph, Brin and Page applied the stochasticity adjustment and primitivity adjustment to convert the original induced matrix H to the irreducible Markov matrix G as follows.

$$G = \alpha H + (1/n)(\alpha a + (1 - \alpha)e)e^T, \tag{5}$$

where n is the number of pages (i.e., states), e is a size-n vector with all ones, a is a size-n vector with $a_i = 1$ if page i is a dangling node (i.e., a node with zero out-degree), and α controls the proportion of time that the random teleportation follows the hyperlinks as opposed to teleporting to a random new page (α is set to be 0.85 in Brin and Page [6]).

3 Leadership discovery

In this section, we first define the problem of leadership discovery.

Problem definition: The problem of leadership discovery is to find the leaders among N synchronized time series, $S^1, S^2, \ldots, S^N$, that exhibit significant lead-lag relations over the set of time series in a real-time manner, where the lead-lag relation is measured by the concept of lagged correlation.

In this work, we are particularly interested in the problem for high-frequency time series, i.e., the time series whose data points are generated in high frequency (seconds, minutes) from the sensors or finance market. Although most of the studies in earth science [27] and financial literature [7, 10] deal with low-frequency data (i.e., weekly, daily), we find that high-frequency data carries more information about lead-lag that is not revealed by low-frequency data. We demonstrate this point by the following real example in financial market.

Figure 2 shows the lagged correlation computed on two stock price time series at four different data frequency (5 seconds, 1 minute, 1 hour and 1 day) when starting from the same date 01/02/2004. As shown in the figure, the low-frequency data (1 day and 1 hour) has a maximum correlation value at zero lag and exhibits positive

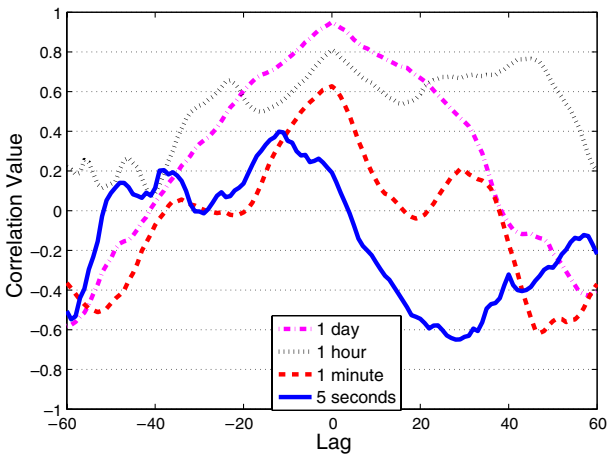


Figure 2 Lagged correlation at different data frequency.

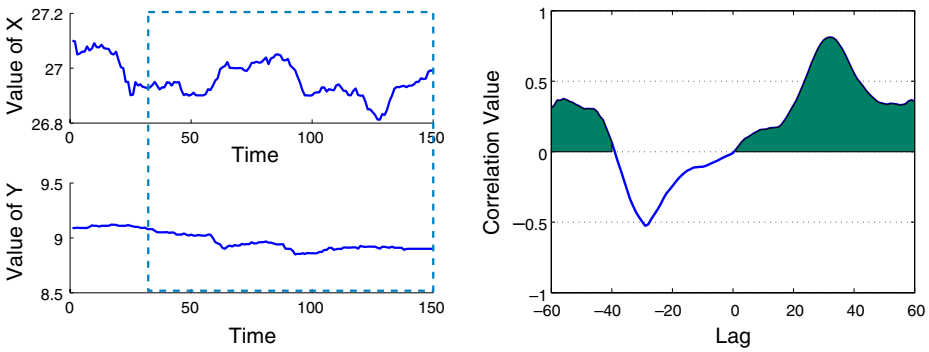
correlation at most of the lags, which implies that the two stocks co-move with the same trend at zero lag from a macro perspective. On the other hand, the correlation computed on high-frequency data (1 minute and 5 seconds) has its peak value at a non-zero lag and exhibits more negative correlations, which means that the two stocks actually undergo a co-movement in an opposite trend with some delay from a micro perspective. Therefore, it is necessary and interesting to investigate the high-frequency data since it indicates the microstructure of the finance market (e.g., the process of practitioners making their trading decision and generating stock price).

Solution overview: Our solution to the problem of leadership discovery has three main steps: (1) compute the lagged correlation between each pair of time series; (2) construct an edge-weighted directed graph based on lagged correlations to analyze the lead-lag relation among the set of time series; (3) detect the leaders by analyzing the leadership transmission in the graph. The preliminary of this work is reported in [32]. We discuss each step in detail.

3.1 Lagged correlation computation

The first step is to compute the lagged correlation between each pair of time series. Existing work [26] on computing lagged correlations cannot be directly applied to our problem, since i) it tries to capture lag correlation in the whole history of streams while our objective is to obtain the local lags in the current sliding window, and ii) the approximation in their updating algorithm has accuracy preference to the points with small lags and may generate a large error for large lags, which is not desirable for our problem. Therefore, we propose to aggregate the effects of various lags and define an *aggregated lagged correlation*. Without loss of generality, we focus on positive correlation, while negative correlation can be handled similarly.

We explain how to compute the aggregated lagged correlation by the following example. Figure 3a shows two time series X (top) and Y (bottom) with a length of 150. The window length is set to be 120 and we consider the window marked by the dotted rectangle. Figure 3b shows the lagged correlation at each lag l computed by (1) over the two windows. The maximum lag $m = 60$, i.e., $|l| \leq 60$. When $l < 0$ (i.e., Y is



(a) Two Time Series (b) Lagged Correlation

Figure 3 Two time series and the lagged correlation plot over their local sliding windows.

delayed), the positive correlation only exists for $l \in [-60, -39]$ (the shadowed area). When $l \geq 0$ (i.e., X is delayed), starting from $l = 1$, we observe a strong increase in positive correlation and it achieves a peak value of 0.81 at $l = 32$. In order to identify the leadership (X leads Y or Y leads X), we need to aggregate all the observed correlation values over the entire lag span and take the expected correlation value given the two cases of l. The aggregated lagged correlation between two time series S^i and S^j, denoted as $E^{ij}(\rho)$, is then defined as the larger expected correlation value:

$$E^{ij}(\rho) = \max\left(E^{ij}(\rho|l \geq 0), E^{ij}(\rho|l < 0)\right). \tag{6}$$

We say that S^i *leads* S^j *if* $E^{ij}(\rho) = E^{ij}(\rho|l < 0)$*, and* S^i *is led by* S^j *otherwise if* $E^{ij}(\rho) = E^{ij}(\rho|l \geq 0)$. Such leadership ($S^i$ leads S^j or vice versa) is also called the *lead-lag* relation between S^i and S^j. The value of $E^{ij}(\rho|l \geq 0)$ is computed as

$$E^{ij}(\rho|l \geq 0) = \sum_{l=0}^{m} \max\left(\rho^{ij}(l), 0\right) \cdot p(l|l \geq 0), \tag{7}$$

where $\max(\rho^{ij}(l), 0)$ takes only positive correlations and $p(l|l \geq 0)$ takes the value of $1/(m+1)$ since the contribution of each lag is equal. $E^{ij}(\rho|l < 0)$ can be computed symmetrically. In Figure 3, by (7), $E^{XY}(\rho|l < 0) = 0.1056$ and $E^{XY}(\rho|l \geq 0) = 0.4017$. Thus, $E^{XY}(\rho) = \max(0.1056, 0.4017) = 0.4017$, indicating that X is led by Y.

3.2 Graph construction

In order to model the leadership relationships among a set of time series, we construct a simple edge-weighted directed graph, $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where the set of nodes $\mathcal{V} = \{S^1, S^2, \ldots, S^N\}$ represents N time series, and the set of directed edges $\mathcal{E}$ represents lead-lag relations between time series. An edge (S^i, S^j) indicates that S^i is led by S^j and its weight is set as $E^{ij}(\rho)$. Since we are interested in significant lead-lag relations, we set a *correlation threshold* γ such that only those pairs S^i and S^j with $E^{ij}(\rho) > \gamma$ have edges in $\mathcal{G}$. It is important to note that, when the window slides, the edges and their weights in $\mathcal{G}$ will change dynamically.

Figure 4 shows the graph $\mathcal{G}$ constructed on 30 component stocks of Dow Jones Industry Average. Each node in the graph represents a stock, each edge represents a significant lagged correlation between two stocks and the arrow on the edge points to the leading stock (we omit the edge weight for clear visualization). For example, there is an edge from Intel Corporation (INTC) to Hewlett-Packard Company (HPQ), which indicates that there is a significant lagged correlation between the world's largest semiconductor company and the largest worldwide seller of personal computers in this sampled period. In the graph, there are some nodes that have relatively more in-links than others, which indicates that these stocks are the leading centers. Citigroup Inc. (C), the major American financial service company, has the largest in-degree of 11, leading 1/3 of Dow Jones component stocks at the time. On the other hand, there are some dangling nodes in the graph, which are not led by any other stocks. Wal-Mart Inc. (WMT), the world's largest public corporation, is an isolated stock without out-link or in-link.

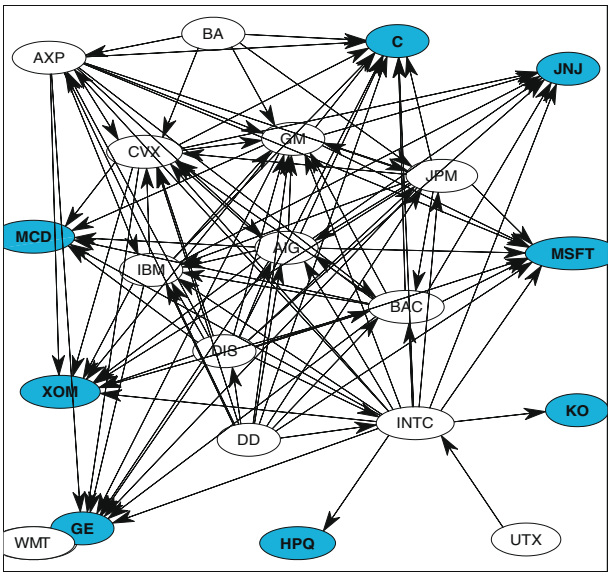


Figure 4 Graph $\mathcal{G}$ on component stocks of dow jones industry average.

3.3 Leader extraction

Given the graph $\mathcal{G}$, we now extract leaders from it. Since a good leader needs to capture both direct and indirect leaderships, we first analyze the leadership transmission in $\mathcal{G}$. Suppose that each time series has a leadership score, based on which a ranking among time series can be obtained. We now discuss how to assign a good leadership score.

Consider the leadership score of A under different graphs as shown in Figure 5. In both Cases I and II, A directly leads three time series, B, C, and D. In Case I, B, C, and D have zero in-degree, while in Case II, C has an in-degree of 3, which implies that A also indirectly leads three time series, E, F, and G. Thus, the leadership score of A in Case II should be larger than that in Case I due to this leadership

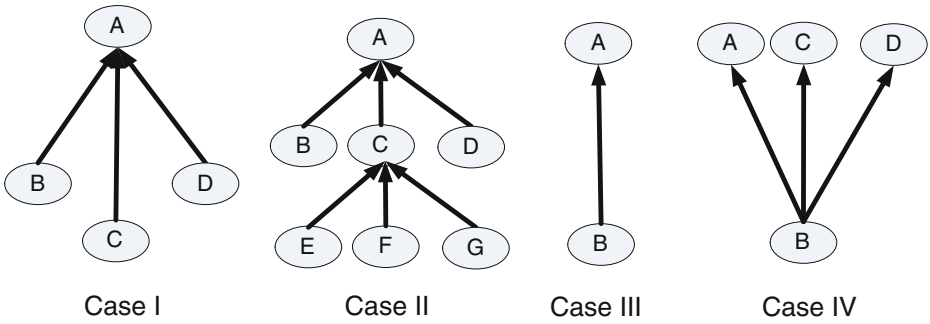


Figure 5 Comparison of leadership score on different graph structures.

transmission. Similarly in Case III, A exclusively leads B, whereas in Case IV, A shares the leadership of B with two time series, C and D. Then the leadership score of A in Case III should be larger than that in Case IV. Therefore, we define the leadership score as

$$score^j = \sum_{S^i \in L_{S^j}} \frac{score^i \, E^{ij}(\rho)}{d_{out}\left(S^i\right)}, \tag{8}$$

where L_{S^j} is the set of time series that are led by S^j, $score^i$ is the leadership score of S^i and $d_{out}(S^i)$ is the summation of the out-edge weights of S^i. The leadership score defined above is similar to the PageRank score defined on the Web Graph to represent the popularity of web pages. In this paper, we adopt PageRank [6] as the leadership score of a time series to quantify its importance in the graph $\mathcal{G}$.

Finally, based on the structure of $\mathcal{G}$ and the PageRank values of time series, we extract the leaders by eliminating redundant leaderships. The basic idea is to first sort the time series by the descending order of their PageRank values and then to remove iteratively the time series that is led either by previously found leaders or by the descendant of the previously found leaders. However, the time series that do not lead any other time series will not be taken as leaders. In addition, when only partial data (i.e., a subset of time series) are correlated, they would form a subgraph that is disconnected from other components of $\mathcal{G}$. By Algorithm 1, we can compute the PageRank scores for the time series in $\mathcal{G}$ including those in this subgraph. Leaders can still be detected from this subgraph since they will not be removed as descendants of other higher-rank leaders in $\mathcal{G}$ due to the disconnection of this subgraph from other parts of $\mathcal{G}$.

3.4 The overall algorithm

Our solution is presented in Algorithm 1. Given the latest values in time series at time t, the algorithm first updates the statistics needed in computing lagged correlations as stated in Section 2. It then computes pairwise aggregated correlations (lines 2–5). Graph $\mathcal{G}$ is then constructed (line 6) and the power method computes the PageRank vector π (line 7). Finally, the *ExtractLeaders* procedure (Algorithm 2) identifies the leaders. In *ExtractLeaders*, time series are first sorted by the descending order of the rank π. Then starting from the time series with the highest rank, it checks the time series led by it and removes them as well as their descendants from the list. The procedure *RemoveDescendant* repeats the process recursively until all descendants of the current leader are removed. The remaining time series on the list are returned as leaders. Note that the set of leaders excludes the time series that are isolated nodes in $\mathcal{G}$ (line 5 in Algorithm 2).

We now analyze the complexity of Algorithm 1. Correlation computation in lines 2–5 needs to compute $(2m+1)N^2$ correlation values, which involves complex mathematical calculation. PageRank computation and the *Extract Leaders* procedure take $\mathcal{O}(kN^2)$ and $\mathcal{O}(N)$ time, respectively, where k is the number of iterations in the power method. Thus, the most time-consuming steps in Algorithm 1 are in computing correlations and PageRank. The space complexity of the algorithm is $\mathcal{O}(mN^2)$ for storing the correlation statistics and $\mathcal{O}(N^2)$ for storing the values in the power method.

Algorithm 1 DiscoverLeaders

INPUT: N time series, $S^1, \ldots, S^N$, up to current time t, sliding window length w, maximum lag m, correlation threshold γ
OUTPUT: *leaders*

Update statistics needed for correlation computation;
for every pair of time series S^i and S^j **do**
  Compute correlation $\rho^{ij}_{t,w}(l)$, for $|l| \leq m$;
  Compute aggregated lagged correlation $E^{ij}(\rho)$ by (6);
end for
Construct graph $\mathcal{G}$ with respect to γ;
Compute PageRank vector π on $\mathcal{G}$;
$L \leftarrow ExtractLeaders(\mathcal{G}, \pi)$;
return L;

Algorithm 2 ExtractLeaders

INPUT: graph $\mathcal{G}$, rank vector π
OUTPUT: *leaders*

$L \leftarrow$ Sort time series in descending order by π;
for each time series S^j in L **do**
  $RemoveDescendant(L, \mathcal{G}, S^j)$;
end for
Remove time series in L with zero indegree in $\mathcal{G}$;
return L;
Procedure $RemoveDescendant(L, \mathcal{G}, S^j)$
for each time series S^i in L after S^j **do**
  if (S^i, S^j) is an edge in $\mathcal{G}$ **then**
    $RemoveDescendant(L, \mathcal{G}, S^i)$;
    Remove S^i from L;
  end if
end for

In a stream environment, correlation computation becomes the bottleneck of Algorithm 1 since the implementation of PageRank is fast when the graph is small enough to be stored in the main memory (e.g., $N = 500$). Too many correlation values need to be computed at each time point and there are endless time points coming into the stream. In order to accomplish prompt leadership detection, we further propose an effective update approach that is able to reduce the number of correlation computations and meanwhile retaining high accuracy, which is described in the next section.

4 Real-time correlation update

In order to speed up the computation of the aggregated lagged correlation for a pair of time series, we propose an efficient update approach by investigating the evolutionary characteristics of lagged correlations. Recall that in (7), all positive

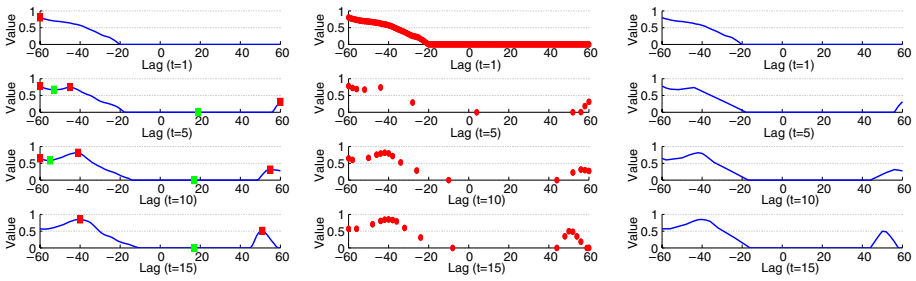
lagged correlation values are aggregated, i.e., we compute the area with positive correlations. Compared with the exact correlation value at each lag, the area formed by these positive correlations is more crucial to determine the lead-lag relation. We call this area the *interesting area*. The basic idea of our update approach is to track the interesting area. More specifically, at an initial time point, we compute the exact correlation value at each lag and record the interesting area. Then at the subsequent time point, we track and update this interesting area by computing the correlation for only a small number of lags. We then use this interesting area to approximate the aggregated lagged correlation.

We now discuss how to track and update the interesting area. Figure 6a gives an example of the evolutionary shapes of the interesting area between two time series. The lagged correlation is computed at each lag $l \in [-60, 60]$. At time $t = 1$, the interesting area spans from $l = -60$ to $l = -20$ and the corresponding correlation value decreases gradually from 0.8 to 0. We call such a continuous area a *wave*. When $t = 5$, we note that there are two waves of the interesting area. The first one spans from $l = -60$ to $l = -17$, which is obviously an evolution from the previous wave. Compared with the wave at $t = 1$, the boundary of this wave enlarges from $l = -20$ to $l = -17$ at $t = 5$. Hereafter, we call this type of wave an *existing wave*. The second wave spans from $l = 55$ to $l = 60$. Since this wave does not exist at $t = 1$, we call it a *new wave*. When $t = 10$ and $t = 15$, the existing wave changes slowly, while the new wave enhances its effect.

The above example shows that, in order to keep track of the interesting area, we need to capture the evolutionary pattern of two types of waves, existing waves and new waves. Our solution is based on two observations.

Observation 1 *An existing wave at time t is relatively stable at subsequent time points after t.*

Observation 1 can be explained as follows. For a specific lag l, the correlation $\rho^{ij}_{t,w}(l)$ at time t is computed on two shifted windows $s^i_{t,w-l}$ and $s^j_{t-l,w-l}$. When the time moves to $t + 1$, correlation $\rho^{ij}_{t+1,w}(l)$ is computed on $s^i_{t+1,w-l}$ and $s^j_{t-l+1,w-l}$. Notice that there is a large overlap in these two sets of windows. Specifically, the difference



(a) Exact Interesting Area (b) Probed Correlation Values (c) Approx. Interesting Area

Figure 6 Tracking the interesting area.

between $s^i_{t,w-l}$ and $s^i_{t+1,w-l}$ (also between the other two windows of S^j) is only one point. As a result, the two correlations $\rho^{ij}_{t,w}(l)$ and $\rho^{ij}_{t+1,w}(l)$ cannot differ a lot. Therefore, we have the above observation of an existing wave.

Using Observation 1, we can track an existing wave as follows. The most important features of a wave are its magnitude and width. The magnitude of a wave can be characterized by its maximum points, while the width can be characterized by the minimum points. Therefore, we propose to approximate the area of an existing wave by tracking its peak points. Specifically, after we compute the exact correlation value for each lag at the initial time point, we record the peak points for the existing wave. Then, at the subsequent time point, we only compute the exact correlation value for the lag of each maximum peak point and conduct a geometric progression probing to both sides of the lag until the probe reaches the boundary. The boundary can be either the adjacent minimum peak point, the maximum lag $\pm m$ or the point with a negative correlation value. Then, we conduct a linear interpolation over the computed correlation points to approximate the area of the wave. Finally, the peak points are updated according to the probed correlation values so that they can be used for the subsequent time point.

Figure 6b shows the points, at which we compute (probe) correlation values. Suppose that $t = 1$ is an initial time point. We compute all the lagged correlation values for $l \in [-60, 60]$ and record a maximum peak point at $l = -60$. When $t = 5$, we probe from the maximum peak point $l = -60$ until reaching the boundary, where we detect a negative correlation. In this process, the probing step is increased exponentially so that the approximated wave has higher accuracy around the peak point. There are altogether 7 correlation values computed in the probing process. Then, as shown in Figure 6c, linear interpolation is applied to these 7 points to form the approximated existing wave. As further shown in $t = 10$ and $t = 15$, this existing wave can be well tracked.

Now, we discuss how to track a new wave. As there is no existent evidence of a new wave at the initial time point, we are not able to record its peaks for the tracking purpose. Fortunately, we have the following observation of new waves.

Observation 2 *A new wave at t only emerges at maximum lag values of* $\pm m$.

Observation 2 can be explained as follows. We first consider the case when $0 \le l \le m$. At a specific time t, the correlation $\rho^{ij}_{t,w}(l)$ is computed on two windows of length $(w - l)$. Therefore, with the increase of l from 0 to m, the window length, on which $\rho^{ij}_{t,w}(l)$ is computed, decreases. On the other hand, compared with the previous time point $t - 1$, each time series evolves by adding a new data point to and deleting an old data point from the sliding window. This causes the value of $\rho^{ij}_{t,w}(l)$ to be different from $\rho^{ij}_{t-1,w}(l)$. However, the effect caused by the new data point is different at different lag l. With the increase of l, the window length becomes smaller and thus the effect of the new data point becomes larger, which results in a larger difference of $\rho^{ij}_{t,w}(l)$ and $\rho^{ij}_{t-1,w}(l)$. This explains why a new wave may emerge at the largest lag $l = m$. Similarly, a new wave is also likely to emerge at $l = -m$.

According to Observation 2, we can track new waves by monitoring the correlation values at $l = \pm m$. As shown in Figure 6b, although there is no sign of a new wave at $l = 60$ when $t = 1$, we also compute its correlation at $t = 5$. This strategy

successfully detects a positive correlation value at $l = 60$. Then, we take it as an existing wave and track it using the approach we have discussed above. Altogether we use 11 points to track the whole interesting area at $t = 5$, saving 91% of correlation computation.

Our update approach, *UpdateCorrelation*, is presented in Algorithm 3. It first checks the correlation values at the two maximum lag points to detect potential new waves (line 2). If there exists a new wave, the algorithm treats it as an existing wave (lines 3–5). Then, the algorithm approximates each existing wave by two procedures *Probe* and *Interpolate* (lines 7–11). Procedure *Probe* is shown in Algorithm 4. After computing the correlation value at the maximum peak point, it probes the points on its two sides in a geometric progression style. The probing stops when the boundary is met, which we have discussed above. As for the procedure *Interpolate*, we use the linear interpolation [20] to connect the probed values and form the approximated

Algorithm 3 UpdateCorrelation

INPUT: new data points at t for two time series S^i and S^j, sliding window length w, maximum lag m, the set of peak points $peak^{ij}_{t-1}$ at time $t-1$
OUTPUT: the lead-lag relation of S^i and S^j

1: **if** there is no existing wave at $l = \pm m$ **then**
2: Compute $\rho^{ij}_{t,w}(m)$ and $\rho^{ij}_{t,w}(-m)$ to detect potential new waves;
3: **if** there exists new waves **then**
4: Add the corresponding l to $peak^{ij}_{t-1}$;
5: **end if**
6: **end if**
7: **for** each maximum peak point $ptMax$ in $peak^{ij}_{t-1}$ **do**
8: $sampleWavePointSet = Probe(ptMax)$;
9: $wavePointSet = Interpolate(sampleWavePointSet)$;
10: Add $wavePointSet$ to corresponding $\rho^{ij}_{t,w}(l)$;
11: **end for**
12: $peak^{ij}_t = detectPeak(\rho^{ij}_{t,w}(l))$;
13: Compute aggregated lagged correlation $E^{ij}(\rho)$ by (6);
14: Decide the lead-lag relation of S^i and S^j;

Algorithm 4 Probe

INPUT: a peak point $ptMax$
OUTPUT: $sampleWavePointSet$

1: $sampleWavePointSet \leftarrow$ Compute $\rho^{ij}_{t,w}(ptMax)$;
2: $step = 1$;
3: $index = ptMax \mp step$; // + for the right-side probing
4: **while** $index$ is not a left/right boundary point **do**
5: $sampleWavePointSet \leftarrow$ Compute $\rho^{ij}_{t,w}(index)$;
6: $step = step \times 2$;
7: $index = ptMax \mp step$; // + for the right-side probing
8: **end while**

interesting area. We then detect and update peak points according to the probed correlation values (line 12), which can be implemented by an existing peak detection algorithm [5]. Finally, we decide the lead-lag relation based on the approximated interesting area (lines 13–14).

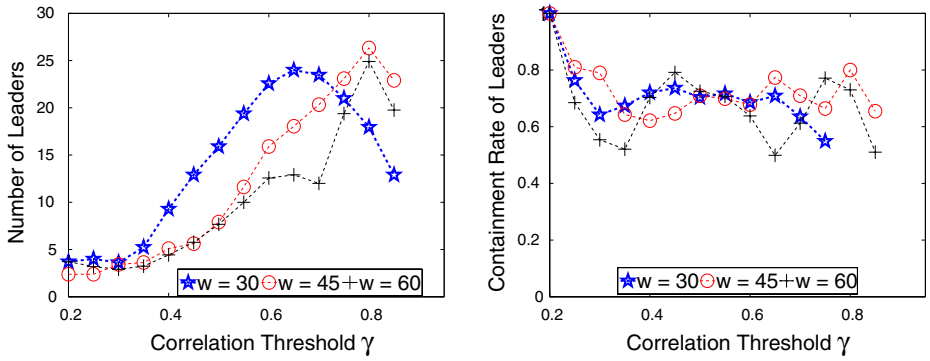
Let us use Figure 6 as an example to illustrate the whole procedure. At the initialization step of time $t = 1$, we compute the correlation value at each lag l for $l \in [-60, 60]$, that is, totally 121 correlations. We also detect peak points for each wave as line 12 of Algorithm 3. In this example, we only record the peak point of $l = -60$ for the wave spanning from $l = -60$ to $l = -20$. We then update the correlation by running Algorithm 3 at subsequent time points. At each time point, we first check whether there are new waves at the two maximum lag points, i.e., $l = \pm 60$ (lines 1–6 of Algorithm 3). As we can see from Figure 6b, at $t = 5$, we detect a new wave at $l = 60$ and record the point of $l = 60$ as a new peak. Then in lines 7–10 of Algorithm 3, we check each of the waves and approximate its shape. We have two waves in this example, one is the existing wave with a previous peak point at $l = -60$ and the other is the newly detected wave with the peak point at $l = 60$. We then start to probe from these two peak points (line 8 of Algorithm 3). The probe method, as described in Algorithm 4, first computes the correlation value of a peak point. It then probes to the two sides respectively by geometric progressing. In this example, for the first wave, it probes from point $l = -60$ to the right until it reaches the boundary point, i.e., the point with the correlation value less than 0. The points we probe are $l = -59, -57, -53, -45, -29, 3$, as shown in Figure 6b. We then interpolate the whole shape using these probed points and record all the correlation values (lines 9–10 of Algorithm 3). The same probing procedure applies to the new wave with peak point at $l = 60$. After that, we get an approximation of the shapes of the two waves. We then update the peak and boundary points and decide the lead-lag relation (lines 12–14 of Algorithm 3). In our example, the existing wave has a larger area than the newly detected wave as shown in Figure 6c.

The *UpdateCorrelation* algorithm enables us to track the interesting area using only $\mathcal{O}(\log m)$ correlation computations instead of $\mathcal{O}(m)$ in a brute-force approach. Moreover, since we start probing from the maximum peak points and stop probing when detecting the boundary, the actual number of correlation computations is much smaller. We further study the efficiency improvement of *UpdateCorrelation* in Section 5.

5 Experimental results

In this section, we design a set of experiments to answer the following questions:

(1) What are the effects of the parameters (e.g., the sliding window length, the correlation threshold) on the performance of our algorithm in terms of discovered leaders?
(2) How does the set of discovered leaders evolve as the sliding window moves forward? Does the set of leaders remain stable or evolve a lot with time?
(3) Are detected leaders interesting and useful? How can we use them?
(4) How effective is *UpdateCorrelation*? How good is its approximation accuracy? Does the accuracy degrade over time?



(a) Number of Leaders (b) Containment Rate of Leaders

Figure 7 Parameter sensitivity on SST 125.

We conduct our experiments on a PC with a Pentium IV 3.4GHz CPU and 2GB RAM and the algorithm is implemented with Matlab. Three real datasets are tested.

- **SST 125**. It contains 125 streams of weekly sea surface temperature on the Pacific ocean from 1990–present.[1] Each stream is normalized using Z-Score [27].
- **S&P 500**. It contains 500 streams of high-frequency stock transaction data which we retrieve from the NYSE Trade and Quote (TAQ) database. We set one tick as 1 minute and extract the data by computing the Volume Weighted Average Price (VWAP) for transactions at each tick as follows:

$$VWAP = \frac{Number\ of\ Share\ Bought \times Share\ Price}{Total\ Share\ Bought}. \tag{9}$$

- **DOW 30**. It contains 30 component stocks of the Dow Jones Industrial Average, which are the largest and most widely held public stocks in the United States. We compute their VWAP by setting one tick as 5 seconds.

5.1 Sensitivity of parameters

There are three parameters in our algorithm: the window length w, the correlation threshold γ and the maximum lag m. As suggested in Box et al. [3], m is set to be $w/2$. Therefore, we only test two paremeters γ and w. We test on 100 consecutive time ticks in SST 125 and vary γ from 0.2 to 0.85 with a step of 0.05. We also test three values of w, i.e., $w = 30$, $w = 45$, and $w = 60$. Figure 7a presents the number of leaders detected at each γ. For all w, we find a clear rise in the number of leaders when γ increases from 0.2 to 0.6. This is because the number of edges in $\mathcal{G}$ decreases with the increase in γ. As $\mathcal{G}$ becomes sparser, the locations are less likely to be covered by the same leader, which results in more leaders. For $w = 30$, when γ exceeds 0.7, there is a drop in the number of leaders. This is because when

[1] http://www.cdc.noaa.gov/data/gridded/. Intraday SST data is available in the same site.

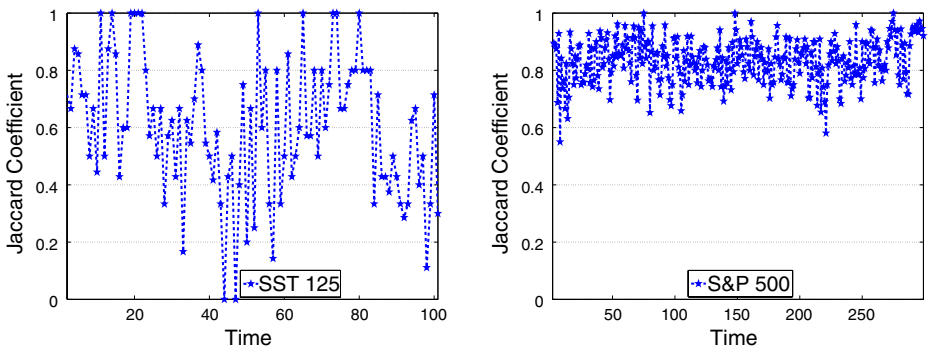
γ is set too high, many locations become isolated and are not led by any others. Therefore, the number of leaders decreases when γ is high and becomes 0 when γ is set as 1, i.e., no edge in $\mathcal{G}$. We also observe similar phenomena for other values of w but with different turning points. In order to study the evolution of leaders when varying γ, we compute the containment rate of leaders between two consecutive γ as $\frac{|Leaders(\gamma_i) \cap Leaders(\gamma_{i-1})|}{|Leaders(\gamma_{i-1})|}$. As shown in Figure 7b, for all w, the containment rate at different γ remains high (averagely 0.7). This indicates that most of the leaders found at a low γ can also be found at a high γ. This gives us a hint in choosing γ. Normally, γ can be set around 0.3 since it tends to select a small number of leaders. If users want to be more confident with the lead-lag relation, γ can be set higher and a higher γ also covers most of the results that are produced by lower ones.

5.2 Stability of leaders over time

A user may raise the following question: since the leaders are updated at every time tick, can I trust the current detected leaders? We now study the stability of leaders over time. We adopt the Jaccard coefficient [28] to measure the similarity between the leaders extracted at two consecutive time ticks, which is computed as $\frac{|Leaders(t_i) \cap Leaders(t_{i-1})|}{|Leaders(t_i) \cup Leaders(t_{i-1})|}$. We test on SST 125 and S&P 500.

For SST 125, we set $w = 30$, $\gamma = 0.3$ and extract leaders at 104 consecutive time ticks in 1997–1998. As shown in Figure 8a, the stability generally remains high (the average similarity is 0.61). The average leader duration (i.e., the time length in which a time series continues to be a leader) is 5.3 ticks (one and a half months) and the maximum duration is 12 ticks (three months). The result suggests that the detected leaders have a certain degree of stability although the interval between two consecutive time ticks is as long as 1 week. Nevertheless, there is a drop of stability in the middle of the Nino phenomena (at around $t = 55$). This is because at $t = 55$, all locations have high anomaly scores as shown in Figure 1. As a result, the lead-lag effect is not significant and the leaders vary from time to time, which results in relatively low leadership stability.

For S&P 500, we set $w = 120$, $\gamma = 0.3$ and extract leaders at 270 consecutive time ticks in an entire trading day. In Figure 8b, we find that the average similarity is as



(a) SST 125 (b) S&P 500

Figure 8 Stability of leaders.

high as 0.82 and is quite stable. This is because its graph $\mathcal{G}$ is large and a small number of altered edges are not likely to affect the PageRank of the stocks. In summary, the results indicate a certain degree of stability for the evolution of the leaders.

5.3 Regular stock leaders

The above experiment for stability of leaders suggests that there exists a set of stocks which act as leaders frequently. We call such leader a *regular leader*. We design a set of experiments to investigate the characteristics of regular leaders. We test the financial datasets of DOW 30 and S&P 500 and count the frequency of each stock being a leader among 100 ticks.

Table 1 gives the top 5 regular leaders in Dow 30 (left part) and S&P 500 (right part). The top 1 regular leaders in these two datasets are AT&T (T) and United Health Group (UNH) with frequency of 47 and 53, respectively, which is around half of the total 100 time ticks. In Figures 9 and 10, we present the rank evolution of the top 5 regular leaders at each time tick for the two datesets, respectively. We can see that there are three types of regular leaders:

(a) The leaders with extremely high ranks in a long continuous time period, such as AT&T (T) in DOW 30 and Dow Chemical (DOW) in S&P 500;
(b) The leaders with steady low ranks, such as Merck (MRK) in DOW 30 and Compuware (CPWR) in S&P 500.
(c) The leaders with jump between high rank and low rank repeatedly, such as Verizon (VZ) in DOW 30 and United Health Group (UNH) in S&P 500.

All of these three types of regular leaders are useful for users. The characteristics of Type-(a) regular leaders suggest that the time series are naturally influential and traded intensively which can lead many other time series constantly. The characteristics of Type-(b) regular leaders indicate that these stocks lead a certain set of other stocks and are less dependent on the market. They tend to be the leaders of corresponding sectors. The characteristics of Type-(c) regular leaders suggest the phenomena that the bursty of new events will trigger the prices of these stock leaders to change first before the market notices and follows. But this lead-lag only occurs occasionally.

5.4 Predictive power

We now demonstrate the usefulness of detected leaders by constructing a Leadership Index, where the weight β_i of each leader in the index portfolio is determined by its relative PageRank value, i.e., $\beta_i = \frac{\pi_i}{\sum_{j \in Leaders} \pi_j}$.

Table 1 Regular leaders on DOW 30 (left) and S&P 500 (right)

Leader	Freq.	Leader	Freq.
AT&T (T)	47	United Health Group (UNH)	53
Verizon (VZ)	40	Alberto-Culver (ACV)	48
Intel (INTC)	35	United States Steel (X)	48
Merck (MRK)	33	Compuware (CPWR)	46
Boeing (BA)	29	Dow Chemical (DOW)	46

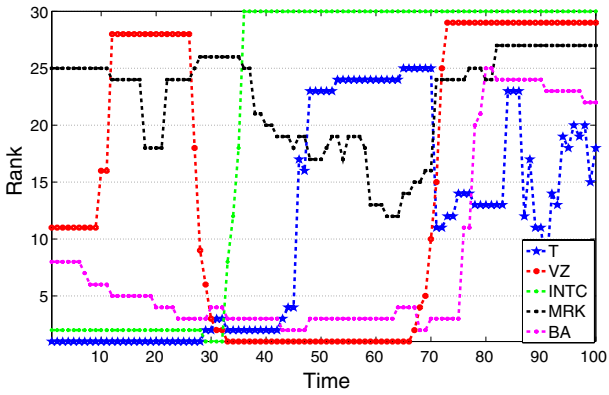


Figure 9 Rank evolution for regular leaders on DOW 30.

Figure 11 presents the Leadership Index on S&P 500. We set $w = 60$ and $\gamma = 0.3$. Among the 500 stocks, we extract an average of 10.8 leaders in a trading day. For comparison, Figure 11 also shows the market index formed by S&P 500. We find that there are five phases in both indices with the upward/downward trend. In the first phase, the two indices rise together with some minor delay in S&P 500 Index. Then, at $t = 95$, the Leadership Index begins to go down first while S&P 500 Index keeps rising until meeting its first turning point at $t = 145$, which is delayed by 50 mins. After that, Leadership Index rebounds at $t = 177$ with a first steady rising trend followed by a steep burst at $t = 209$. In contrast, S&P 500 Index starts the rising trend at $t = 197$ and meets the burst point at $t = 214$, which are both delayed with Leadership Index. The final turning point of S&P 500 Index is at $t = 233$, which is delayed with Leadership Index by 7 mins. In summary, in the first phase, Leadership Index leads S&P 500 Index with very small lags; while in other phases, Leadership Index leads S&P index with larger lags and the lag decreases from 50 mins at the beginning to 7 mins at the end. We conduct Granger-causality analysis over these two indices and the result suggests that Leadership Index Granger-causes S&P 500 index where the optimal lagged value is 1 for both indices with a significant F-Statistics of 9.65.

We find similar results in SST 125 dataset. Recall that in Figure 1, at the beginning and the ending of Nino phenomena, Leadership Index leads Nino 1+2 index with

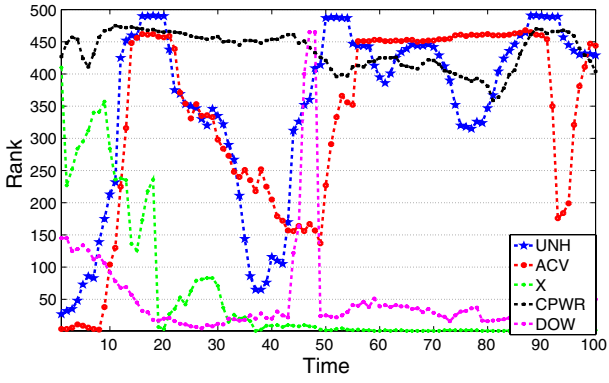


Figure 10 Rank evolution for regular leaders on S&P 500.

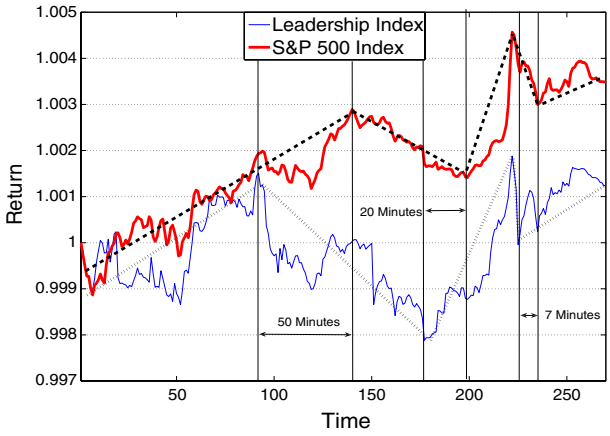


Figure 11 Leadership index v.s. market index on S&P 500.

large lags, whilst in the middle phase of the phenomena, the lead-lag effect is not so significant with small lags.

The above findings indicate that the leadership index indeed exhibits a predictive ability. However, its predictive power has different strengths at different time. Then, how can we know the predictive strength of the Leadership Index at a specific point of time? We study again the shape of the *interesting area* and differentiate two types of waves, the zero-lag wave and the non-zero-lag wave. The zero-lag wave is centered around the lag value of 0. Two time series having a zero-lag wave tend to have a low predictive power due to the small lag. On the other hand, a non-zero-lag wave indicates a large time lag, which is the cause of the high predictive power. We define the strength of zero-lag correlations as the fraction of the edges in $\mathcal{G}$ that have zero-lag waves. The strength indicates the extent that the graph $\mathcal{G}$ is contributed by zero-lag correlations. Therefore, a low zero-lag correlation strength indicates a high predictive power and vice versa.

Figure 12 presents the zero-lag correlation strength over time on the two datasets SST 125 and S&P 500. We find that the strength for SST 125 is low at the beginning when the Nino phenomena starts to emerge. After the Nino phenomena develops fully, all the locations tend to have synchronized anomalies and the strength becomes high as 0.7. Finally, when the phenomena begins to diminish, some locations lead others to drop and the strength falls down again, which results in the increase of predictive power. As for S&P 500, we observe a high but decreasing strength curve starting from $t = 1$ and it reaches 0.1 at $t = 95$ (matching with the end of the first rising phase of Leadership Index in Figure 11). It then stays very low below 0.2 until the end of the trading day. Therefore, the evolution pattern of the zero-lag strength coincides with the change of the predictive power of Leadership Index.

5.5 Indegree vs. pagerank

This set of experiments compare two methodologies for time series ranking, by the value of indegree and by the value of PageRank in the graph $\mathcal{G}$. At each time tick, we use the value of indegree to rank each time series and use Algorithm 2 to extract leaders. Then the Indegree Index is constructed where the weight of each leader is the proportion of their indegree value in the total indegree values of all the leaders.

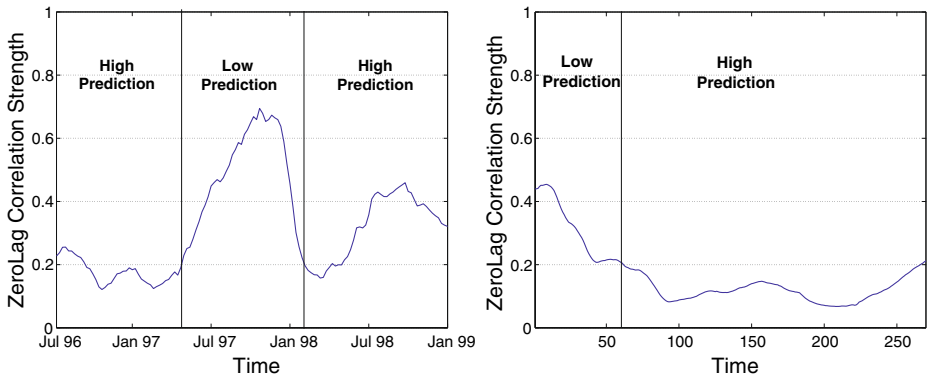
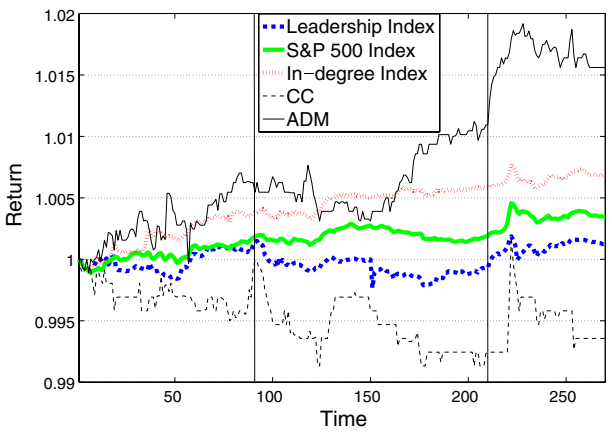


Figure 12 Zero-lag strength of leaders.

Figure 13 presents the Indegree Index on S&P 500. Compared with the Leadership Index, the Indegree Index does not show any predictive power. We first study the turning point of the time series. At $t = 95$, Leadership Index begins to go down while Indegree Index keeps rising. We further check the set of leaders extracted by these two approaches. For Leadership Index, the top-1 leader is Circuit City Stores, Inc. (CC), which constitutes 0.42 of the portfolio. From $t = 95$ to $t = 96$, its return drops from 1 to 0.9987 and leads to a loss of 0.6% of the index, which is 79.6% of the total loss of the index. After that, its return keeps dropping and reaches 0.997 in 5 mins and during this period, it is always a constitution of Leadership Index. As for Indegree Index, at $t = 95$, CC is not taken as leaders as its indegree is as low as 49 and ranked 409 by the indegree value. The top-1 leader of Indegree Index is Qwest Communications International Inc. (Q), which rises 0.1% from $t = 95$ to $t = 96$ and leads 248 other time series. However, it is only ranked 21 by PageRank and is not taken as leaders by Leadership Index since it is led by CC. Similar situations also happen at other turning points. For example, at $t = 209$ where Leadership Index

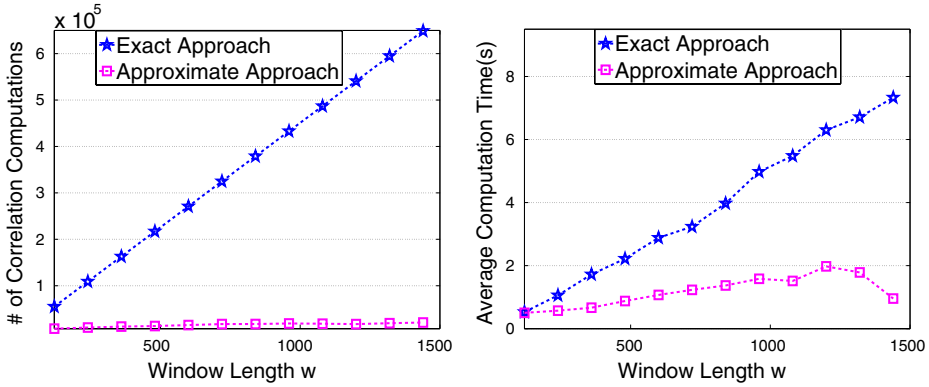
Figure 13 Indegree index v.s. leadership index on S&P 500.



starts a burst of rise, the largest return is contributed by the leader, Archer Daniels Midland Company (ADM), which increases 0.21% of the return (60% of the total return in the portfolio). However, it only leads 49 stocks (ranked 374 by Indegree) and is not taken as a constitution in Indegree Index. This phenomena reveals the fact that the stocks with the highest in-degree are not necessarily the leaders of the whole market. They could just be some intermediate nodes between the leaders and other stocks. Therefore, the time lag between them and the stocks they lead is not large enough to exhibit obvious prediction power. Instead, they tend to have stronger zero-lag tracking power, that is, they usually evolve synchronically with the market index. As shown in Figure 13, the trend of Indegree Index matches perfectly to that of S&P 500 Index. In contrast, the PageRank method considers the transition property of the lead-lag effect. The stocks directly led by intermediate stocks tend to have larger lags to the leaders than to the intermediate stocks. Therefore, the leaders in Leadership Index show stronger prediction power.

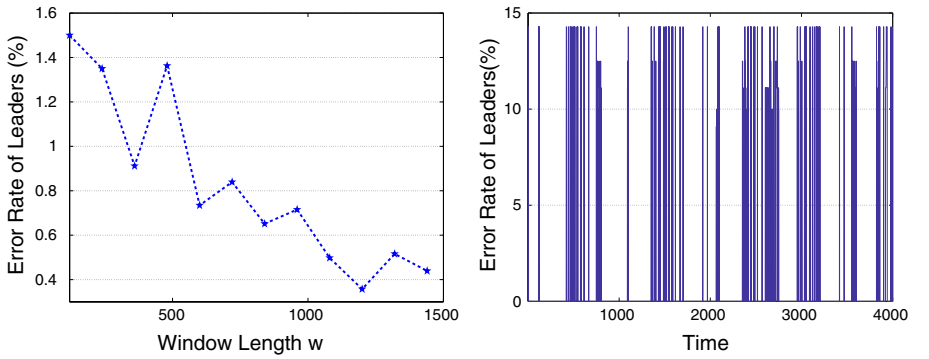
5.6 Correlation update

We now study the effectiveness of the *UpdateCorrelation* algorithm. We test on DOW 30 and vary w from 120 to 1,440. For each w, we move forward the sliding window over that trading day and compare our approximate approach with the exact approach. Figure 14a reports the number of correlation computations. When $w = 120$, the exact approach needs around 54,000 correlation computations, while our approximate approach only needs 7,571 computations. The number of correlation computations for the exact approach increases linearly with w, while our approximate approach grows very slightly with w. When $w = 1{,}440$, our approximate approach needs to compute 20,767 correlation values, which is over 30 times less than 648,000 computations of the exact approach. Figure 14b presents the average running time for the two approaches, which shares a similar trend with the correlation computations in Figure 14a. When $w = 1{,}440$, the running time for approximate approach is 0.94 s, which is an order of magnitude faster than 9.3 s of the exact approach.



(a) Number of Correlation Computations (b) Running Time

Figure 14 Efficiency of correlation update on DOW 30.



(a) Approximation Error Rate (b) Approximation Error Rate Over Time

Figure 15 Accuracy of correlation update on DOW 30.

Figure 15a shows the accuracy of the approximation. The error rate is computed as the Jaccard distance between the two sets of leaders detected by the two approaches. And the average error rate is less than 1.5% and decreases when w increases. Figure 15b also presents the approximation error rate over time when we move forward the sliding window by setting $w = 360$ and $\gamma = 0.3$. It shows that the error is always lower than 15% as time goes far away from the initial time tick. This justifies that our approximate approach is able to refine peak values and achieve good approximation accuracy.

6 Related work

There are several existing studies on multiple time series stream mining [18, 22, 27, 30]. Papadimitriou et al. [22] tracked local correlations by comparing the local auto-covariance matrices of each time series. Zhu and Shasha [33] monitored thousands of time series data but focused on finding high cross-correlation pairs of them. Kontaki et al. [18] used the concept of subspace α-cluster to cluster streaming time series. Steinbach et al. [27] analyzed the linear correlation of multiple climate time series and attempted to construct climate index using clustering. Sakurai et al. [26] proposed an algorithm named BRAID to detect arbitrary lag correlations among time series. BRAID uses a geometric probing strategy and sequence smoothing to approximate the lag value wave. Since BRAID always starts probing from lag $l = 0$, the approximation generates larger error when l becomes larger. In our work, on the contrary, we track features of each interesting area, i.e., the peaks and boundaries, and probe from each local maximum peaks. This gives a good approximation accuracy for the wave at large l. To the best of our knowledge, our work is the first to discover the leadership among multiple time series.

We are also aware of a stream of work [10, 16, 17, 31] that constructs a weighted graph on time series in order to discover different interesting patterns. Dorr and Denton [10] proposed to discover timing patterns that begin earlier, end later, and are longer among/than the time series of the same pattern conglomerate, which did not consider lead-lag effect when aligning two time series. Wichard et al. [31] utilized

"mixed state analysis" to capture the linear interrelation between the dynamics of stock prices' return. Idé and Kashima [16] proposed an anomaly detection method by analyzing the eigenspace of the dependency matrix. Later, Idé et al. [17] computed the anomaly score of a time series by investigating its k-neighborhood time series. Different from these studies, our work discovers leaders by constructing a graph based on the lead-lag relations of time series.

There are also some existing studies on object ranking in a graph, including PageRank for ranking web pages/documents [6, 13], PopRank for ranking web objects (e.g., products, publications, people) [21], and a mechanism for ranking news articles and news sources [9]. Different from these studies, our work is to rank time series and detect leaders based on the ranking.

Finally, we review some empirical studies on lead-lag effect in different domains. In world wide web, Gruhl et al. [15] studied the information diffusion along topics and individuals through blogspace. Gruhl et al. [14] further detected the lead-lag relation between online chatter and the peak of sales volume. There are similar studies and theories in high-frequency finance. One of the theoretical analysis was based on the study of informativeness [25]. If the price of a security is informative for prices of other securities, its return will lead those of other securities. The inside mechanism is that trading reveals information that causes price revisions of securities with correlated underline values or information. Other empirical work [4] concentrated on the speed of price adjustment. In this mechanism, a security is said to lead other securities if its price adjustment to a common factor is earlier than that of other securities. These two mechanisms are the causes of the lead-lag effect in high-frequency finance domain and our approach is able to detect the phenomena caused by them efficiently.

7 Conclusions

In this paper, we formalize a novel problem of discovering leaders from multiple time series based on lagged correlation. A time series is identified as a leader if its movement triggers the co-movement of many other time series. We develop an efficient algorithm to detect leaders in a real-time manner. The experiments on real earth science data and financial data show that the discovered leaders demonstrate high predictive power on the event of general time series entities and the approximate correlation update approach is up to an order of magnitude faster than the exact approach at a relative low error rate.

Acknowledgements The work was supported by the grants of the Research Grants Council of the Hong Kong SAR, CUHK No. 419008 and 419109, and the Chinese University of Hong Kong Direct Grant No. 2050474.

References

1. Langville, A.N., Meyer, C.D.: Google's PageRank and Beyond: The Science of Search Engine Rankings. Princeton University Press (2006)
2. Bhuyan, R.: Information, alternative markets, and security price processes: a survey of literature. Finance 0211002, EconWPA (2002)

3. Box, G., Jenkins, G.M., Reinsel, G.: Time Series Analysis: Forecasting and Control. Prentice Hall (1994)
4. Brennan, M.J., Jegadeesh, N., Swaminathan, B.: Investment analysis and the adjustment of stock prices to common information. Rev. Financ. Stud. **6**(4), 799–824 (1993)
5. Brent, R.P.: Algorithms for Minimization Without Derivatives. Dover Publications (2002)
6. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Comput. Netw. ISDN Syst. **30**(1–7), 107–117 (1998)
7. Campbell, J.Y., Grossman, S.J., Wang, J.: Trading volume and serial correlation in stock returns. Q. J. Econ. **108**(4), 905–939 (1993)
8. Chan, K.: A further analysis of the lead-lag relationship between the cash market and stock index futures market. Rev. Financ. Stud. **5**(1), 123–152 (1992)
9. Corso, G.M.D., Gullí, A., Romani, F.: Ranking a stream of news. In: WWW '05: Proceedings of the 14th International Conference on World Wide Web, pp. 97–106. ACM, New York (2005)
10. Dorr, D.H., Denton, A.M.: Establishing relationships among patterns in stock market data. In: Data & Knowledge Engineering (2008)
11. Douglis, F., Ball, T., Chen, Y.-F., Koutsofios, E.: The AT&T internet difference engine: tracking and viewing changes on the web. World Wide Web **1**(1), 27–44 (1998)
12. Granger, C.W.J.: Investigating causal relations by econometric models and cross-spectral methods. Econometrica **37**(3), 424–38 (1969)
13. Greco, G., Greco, S., Zumpano, E.: A probabilistic approach for distillation and ranking of web pages. World Wide Web **4**(3), 189–207 (2001)
14. Gruhl, D., Guha, R., Kumar, R., Novak, J., Tomkins, A.: The predictive power of online chatter. In: KDD, pp. 78–87. ACM, New York (2005)
15. Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. In: WWW, pp. 491–501. ACM, New York (2004)
16. Idé, T., Kashima, H.: Eigenspace-based anomaly detection in computer systems. In: KDD, pp. 440–449 (2004)
17. Idé, T., Papadimitriou, S., Vlachos, M.: Computing correlation anomaly scores using stochastic nearest neighbors. In: ICDM, pp. 523–528
18. Kontaki, M., Papadopoulos, A.N., Manolopoulos, Y.: Continuous subspace clustering in streaming time series. Inf. Syst. **33**(2), 240–260 (2008)
19. Kumar, R., Novak, J., Raghavan, P., Tomkins, A.: On the bursty evolution of blogspace. World Wide Web **8**(2), 159–178 (2005)
20. Meijering, E.: Chronology of interpolation: From ancient astronomy to modern signal and image processing. In: Proc. of the IEEE, pp. 319–342 (2002)
21. Nie, Z., Zhang, Y., Wen, J.-R., Ma, W.-Y.: Object-level ranking: bringing order to web objects. In: WWW, pp. 567–574 (2005)
22. Papadimitriou, S., Sun, J., Yu, P.S.: Local correlation tracking in time series. In: ICDM, pp. 456–465 (2006)
23. Pirolli, P., Pitkow, J.E.: Distributions of surfers' paths through the world wide web: Empirical characterizations. World Wide Web **2**(1–2), 29–45 (1999)
24. Pitkow, J.E.: Summary of www characterizations. World Wide Web **2**(1–2), 3–13 (1999)
25. Säfvenblad, P.: Lead-lag effects when prices reveal cross-security information. Working Paper Series in Economics and Finance 189. Stockholm School of Economics (1997)
26. Sakurai, Y., Papadimitriou, S., Faloutsos, C.: Braid: stream mining through group lag correlations. In: SIGMOD, pp. 599–610 (2005)
27. Steinbach, M., Tan, P.-N., Kumar, V., Klooster, S.A., Potter, C.: Discovery of climate indices using clustering. In: KDD, pp. 446–455 (2003)
28. Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison-Wesley (2006)
29. von Storch, H., Zwiers, F.W.: Statistical Analysis in Climate Research. Cambridge University Press (2002)
30. Wang, Q., Megalooikonomou, V.: A dimensionality reduction technique for efficient time series similarity analysis. Inf. Syst. **33**(1), 115–132 (2008)
31. Wichard, J.D., Merkwirth, C., Ogorzałlek, M.: Detecting correlation in stock market. Physica, A **344**(1–2), 308–311 (2004)
32. Wu, D., Ke, Y., Yu, J.X., Yu, P.S., Chen, L.: Detecting leaders from correlated time series. In: DASFAA, pp. 352–367 (2010)
33. Zhu, Y., Shasha, D.: Statstream: statistical monitoring of thousands of data streams in real time. In: VLDB, pp. 358–369 (2002)