Detecting Leaders from Correlated Time Series

Di Wu[1], Yiping Ke[1], Jeffrey Xu Yu[1], Philip S. Yu[2], and Lei Chen[3]

[1] The Chinese University of Hong Kong
{dwu, ypke, yu}@se.cuhk.edu.hk
[2] University of Illinois at Chicago
psyu@cs.uic.edu
[3] The Hong Kong University of Science and Technology
leichen@cse.ust.hk

Abstract. Analyzing the relationships of time series is an important problem for many applications, including climate monitoring, stock investment, traffic control, etc. Existing research mainly focuses on studying the relationship between a pair of time series. In this paper, we study the problem of discovering leaders among a set of time series 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 as time goes by, since the dynamic nature of time series results in highly evolving relationships between time series. 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 climate science data and stock data show that our algorithm is able to compute time series leaders efficiently in a real-time manner and the detected leaders demonstrate high predictive power on the event of general time series entities, which can enlighten both climate monitoring and financial risk control.

1 Introduction

In the literature, the lagged correlation between two streams has been well studied in empirical research [5, 1, 12] and efficient algorithms to discover lagged correlations have also been developed [13]. 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. Two application examples are given as follows.

Earth Science: In climate teleconnection network, each stream represents the weather observations (e.g., temperature, pressure and precipitation) [16] 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

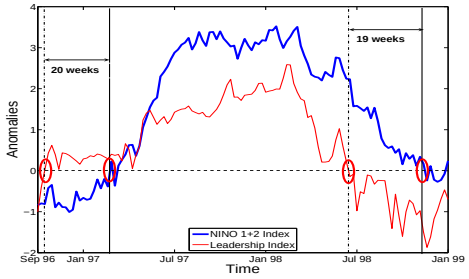


Fig. 1. Leadership Index VS. NINO 1+2 Index from 1996-1999

earth scientists can understand better from which location the climate 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 [12]. 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.

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 climate dataset. We analyze the streams of the sea surface temperature (SST) on the Pacific ocean ($30^\circ S - 30^\circ N$, $55^\circ E - 80^\circ W$) where the famous Nino phenomena occurs irregularly every 4-5 years. We study a period from 1996-1999. In Fig. 1, the bold blue line shows the weekly NINO 1+2 index which is a standard climate 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 Fig. 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 [7] 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 Fig. 1. Therefore, analyzing leaders can detect the trend of an event at an early stage. In climate observation and control, this predictive power is very helpful in giving the scientists an early alert on the climate phenomena and allowing them to do better preventions for the coming disasters.

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.

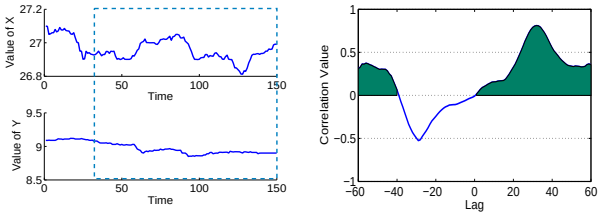
In this paper, we propose an efficient streaming algorithm to address the problem. The main contributions of the paper are summarized as follows. First, we formalize a new problem of discovering the leadership among multiple time series, which well captures the overall co-movements of time series. Second, 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 climate 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 $\{S^1, S^2, \ldots, S^N\}$, 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 is the value of S^j at the most recent time point t. Given a length w and a time point t, a sliding window for time series S^j, denoted as $s_{t,w}^j$, is the subsequence $(s_{t-w+1}^j, \ldots, s_t^j)$. And 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:

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



(a) Two Time Series (b) Lagged Correlation

Fig. 2. Two Time Series and the Lagged Correlation Plot over their Local Sliding Windows

where $\overline{s^i_{t,w-l}}$ and $\overline{s^j_{t-l,w-l}}$ are the mean values in the shifted sliding windows $s^i_{t,w-l}$ and $s^j_{t-l,w-l}$, and $\sigma^i_{t,w-l}$ and $\sigma^j_{t-l,w-l}$ are the standard deviations. In particular, $\rho^{ij}_{t,w}(0)$ is the correlation with zero lag (known as the local Pearson's correlation [11]). When $l > 0$, $\rho^{ij}_{t,w}(l)$ denotes the correlation between the sliding windows $s^i_{t,w}$ and $s^j_{t,w}$ by delaying S^i with a lag l. The case when $l < 0$ can be easily handled symmetrically. Since $\rho^{ij}_{t,w}(l)$ is computed on the common parts of two windows, l is less than the window length w, and in practice $|l| \leq w/2$ as suggested in [2]. 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 [18, 13], the lagged correlation can be computed efficiently by tracking the following statistics: the inner product, the sum of squares and the sum of the shifted windows $s^i_{t,w-l}$ and $s^j_{t-l,w-l}$.

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.

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. We now 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 [13] 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. Fig. 2(a) 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. Fig. 2(b) shows the lagged correlation at each lag l computed by Eq. (1) over the two windows. The maximum lag $m = 60$, i.e., $|l| \leq 60$. When $l < 0$ (i.e., Y is 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:

$$E^{ij}(\rho) = \max(E^{ij}(\rho|l \geq 0), E^{ij}(\rho|l < 0)). \tag{2}$$

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

$$E^{ij}(\rho|l \geq 0) = \sum_{l=0}^{m} \max(\rho^{ij}(l), 0) \cdot p(l|l \geq 0), \tag{3}$$

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 Fig. 2, by Eq. (3), $E^{XY}(\rho|l < 0) = 0.1056$ and $E^{XY}(\rho|l \geq 0) = 0.4017$. Thus, $E^{XY}(\rho) = max(0.1056, 0.4017) = 0.4017$ indicating 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.

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 Fig. 3. In case I and II, A directly leads 3 time series, B, C, and D. In case I, all of the three have zero

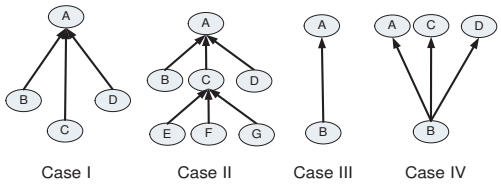


Fig. 3. Comparison of Leadership Score on Different Graph Structures

in-degree. In case II, *C* has an in-degree of 3, which implies that *A* indirectly leads the three that are led by *C* as well as the three directly led by *A* itself. It indicates that the leadership score of *A* in case II should be larger than that in case I. On the other hand, consider case III and case IV. In case III, *B* is exclusively led by *A*, whereas in case IV, *B* is led by *A* as well as the other two, *C* and *D*. The leadership score of *A* in case III should be larger than that in case IV. Therefore, we define leadership score as

$$score^j = \sum_{S^i \in L_{Sj}} \frac{score^i E^{ij}(\rho)}{d_{out}(S^i)}, \tag{4}$$

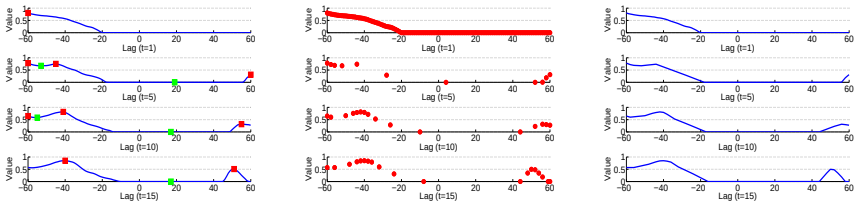
where L_{Sj} is the set of time series that are led by S^j, $score^i$ is the leadership score of S^i and $d_{out}(S^i)$ is the summation of out edge weights of S^i. This leadership score defined above is similar to that defined for the Web Graph on which PageRank score is computed to represent the popularity of web pages. In this paper, we adopt PageRank [4] 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 previously found leaders.

3.4 The Overall Algorithm

Our solution is presented in Algorithm 1. Given the latest values in time series at time point 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 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.

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 *ExtractLeaders* procedure take $O(kN^2)$ and $O(N)$ time, respectively, where k is the number of power method iterations. Thus, the most time-consuming steps in Algorithm 1 are in computing correlations and



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

Fig. 4. Tracking the Interesting Area

PageRank. The space complexity of the algorithm is $O(mN^2)$ for storing the correlation statistics and $O(N^2)$ for storing the values in 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*

1: Update statistics needed for correlation computation;
2: **for** every pair of time series S^i and S^j **do**
3: Compute correlation $\rho^{ij}_{t,w}(l)$, for $|l| \leq m$;
4: Compute aggregated lagged correlation $E^{ij}(\rho)$ by Eq. (2);
5: **end for**
6: Construct graph $\mathcal{G}$ with respect to γ;
7: Compute PageRank vector π on $\mathcal{G}$;
8: $L \leftarrow ExtractLeaders(\mathcal{G},\pi)$;
9: **return** L;

Algorithm 2 ExtractLeaders

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

1: $L \leftarrow$ Sort time series in descending order by π;
2: **for** each time series S^j in L **do**
3: *RemoveDescendant*$(L,\mathcal{G},S^j)$;
4: **end for**
5: **return** L;
6: **Procedure** *RemoveDescendant*$(L,\mathcal{G},S^j)$
7: **for** each time series S^i in L after S^j **do**
8: **if** (S^i,S^j) is an edge in $\mathcal{G}$ **then**
9: *RemoveDescendant*$(L,\mathcal{G},S^i)$;
10: Remove S^i from L;
11: **end if**
12: **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 store 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 following 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 Eq. (3), all positive lagged correlation values are aggregated, i.e., we compute the area with positive correlations. Therefore, 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. Fig. 4(a) 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 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 $l = -17$. 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 this type of wave a *new wave*. When $t = 10$ and $t = 15$, the existing wave changes slowly, while this 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 between $s^i_{t,w-l}$ and $s^i_{t+1,w-l}$ (also between the other two windows) 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.

Fig. 4(b) 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 Fig. 4(c), 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, the remaining problem is 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 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 $0 \leq l \leq m$. At a specific time t, the correlation $\rho_{t,w}^{ij}(l)$ is computed on two windows of length $(w-l)$. Therefore, with the increase of l from 0 to m, the window length, on which $\rho_{t,w}^{ij}(l)$ 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 $\rho_{t,w}^{ij}(l)$ to be different from $\rho_{t-1,w}^{ij}(l)$. However, the effect of the time series evolvement on the value of $\rho_{t,w}^{ij}(l)$ is different for different lag l. With the increase of l, the windows, on which $\rho_{t,w}^{ij}(l)$ is computed, becomes smaller and thus the effect of the evolvement becomes larger, which results in larger difference of $\rho_{t,w}^{ij}(l)$ and $\rho_{t-1,w}^{ij}(l)$. 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 Fig. 4(b), 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. In summary, at $t = 5$, we use 11 points to track the whole interesting area, 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 *Interpo* (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 *Interpo*, we use the linear interpolation [10] to connect the probed values and form the approximated 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 [3]. Finally, we decide the lead-lag relation based on the approximated interesting area (Lines 13-14).

The *UpdateCorrelation* algorithm enables us to track the interesting area using only $O(\log m)$ correlation computations instead of $O(m)$ that a brute-force approach requires. Moreover, since we start probing from the maximum peak points and stop probing when

Algorithm 3 UpdateCorrelation

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

if there is no existing wave at $l = \pm m$ **then**
  Compute $\rho_{t,w}^{ij}(m)$ and $\rho_{t,w}^{ij}(-m)$ to detect potential new waves;
  if there exists new waves **then**
    Add the corresponding l to $peak_{t-1}^{ij}$;
  end if
end if
for each maximum peak point $ptMax$ in $peak_{t-1}^{ij}$ **do**
  $sampleWavePointSet = Probe(ptMax)$;
  $wavePointSet = Interpo(sampleWavePointSet)$;
  Add $wavePointSet$ to corresponding $\rho_{t,w}^{ij}(l)$;
end for
$peak_t^{ij} = detectPeak(\rho_{t,w}^{ij}(l))$;
Compute aggregated lagged correlation $E^{ij}(\rho)$ by Eq. (2);
Decide the lead-lag relation of S^i and S^j;

Algorithm 4 Probe

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

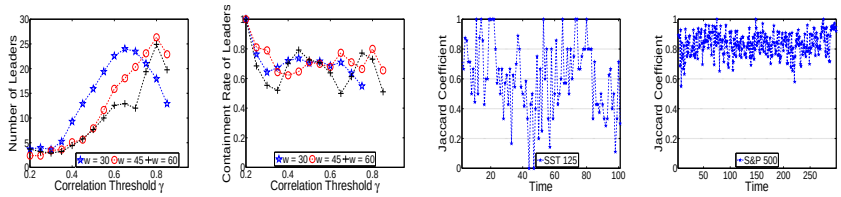
$sampleWavePointSet \leftarrow$ Compute $\rho_{t,w}^{ij}(ptMax)$;
$step = 1$;
$index = ptMax \mp step$; // + for right side probe
while $index$ is not a left(right) boundary point **do**
  $sampleWavePointSet \leftarrow$ Compute $\rho_{t,w}^{ij}(index)$;
  $step = step \times 2$;
  $index = ptMax \mp step$; // + for right side probe
end while

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 appropriately?
(4) How effective is *UpdateCorrelation*? How good is its approximation accuracy? Does the accuracy degrade over time?



(a) Number of Leaders in SST 125 (b) Containment Rate of Leaders in SST 125 (c) Stability(SST) (d) Stability(S&P)

Fig. 5. Parameter Sensitivity and Leaders Stability

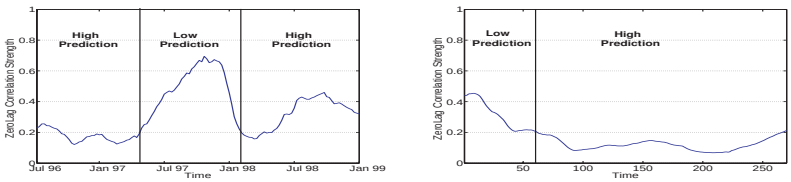
We perform our experiments on a PC with a Pentium IV 3.4GHz CPU and 2GB RAM and the algorithm is implemented with Matlab. We test by using two real datasets.

- **SST 125**. It contains 125 streams of weekly sea surface temperature on the Pacific ocean from 1990-present[4]. Each stream is normalized using Z-Score [14].
- **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 extract the tick data of stock prices by computing the Volume Weighted Average Price (VWAP) for transactions at each tick as $VWAP = \frac{Numberof ShareBought \times SharePrice}{TotalShareBought}$.

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 [2], 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 $w = 30, 45, 60$. Fig. 5(a) 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 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 Fig. 5(b), 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.

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 [15] to measure the similarity between the leaders extracted at two consecutive time ticks, which is

[4] http://www.cdc.noaa.gov/data/gridded/



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

Fig. 6. Zero-Lag Strength of Leaders

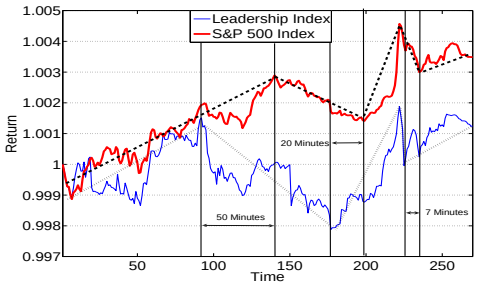
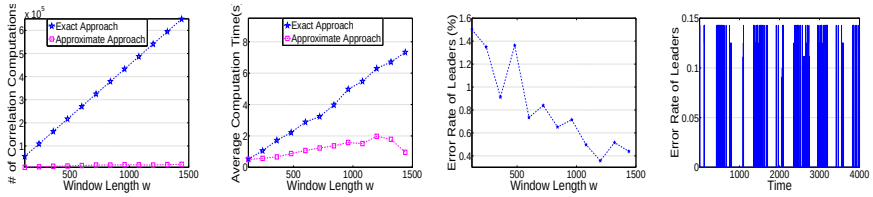


Fig. 7. Leadership Index VS. S&P 500 Market Index

computed as $\frac{|Leaders(t_i) \cap Leaders(t_{i-1})|}{|Leaders(t_i) \cup Leaders(t_{i-1})|}$. For SST 125, we set $w = 30$, $\gamma = 0.3$ and extract leaders at 104 consecutive time ticks in 1997-1998. As shown in Fig. 5(c), the stability generally remains high (the average similarity is 0.61). The average leader duration (i.e., the time length in which a stock 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(around $t = 55$). This is because all locations have high anomaly scores as shown in Fig. 1 at that time. Therefore, 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 Fig. 5(d), we find that the average similarity is 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 stocks' PageRank. In summary, the results indicate a certain degree of stability for the evolution of the leaders.

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., $\beta_i = \frac{\pi_i}{\sum_{j \in Leaders} \pi_j}$. Fig. 7 presents the Leadership Index on S&P 500. We extract 1-minute interval data and set $w = 60$, $\gamma = 0.3$. Among the 500 stocks, we extract an average of 10.8 leaders in a trading day. Compared with the market index formed of S&P 500, we find there are five phases in both indices with the upward/downward trend. In the first phase, these 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 meets its first turning point at $t = 145$, which is delayed by 50 minutes. After that, Leadership



(a) Number of Correlation Computations (b) Running Time (c) Approximation Error Rate (d) Approximation Error Rate Over Time

Fig. 8. Performance of Correlation Update

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 minutes. 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 minutes at the beginning to 7 minutes 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 Fig. 1, at the beginning and the ending of Nino phenomena, Leadership Index leads Nino 1+2 index with 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. Fig. 5 presents the zero-lag correlation strength over time on the two datasets SST 125 and S&P500. 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. On the other hand, 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 Fig. 7). 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.

Correlation Update: We now study the effectiveness of the *UpdateCorrelation* algorithm. In order to have a longer and consistent time series to test, we extract 30 stocks with tick frequency of 5 seconds and vary w from 120 to 1440. For each w, we move forward the sliding window over that trading day and compare our approximate approach with the exact approach. Fig. 8(a) reports the number of correlation computations. When $w = 120$, the exact approach needs around $54,000$ correlation computations, while our approximate approach only needs 7571 computations. The number of correlation computations for the exact approach increases linearly with w, while our approximate approach grows very slowly with w. When $w = 1440$, our approximate approach needs to compute 20,767 correlation values, which is over 30 times less than $648,000$ computations of exact approach. Fig. 8(b) presents the average running time for the two approaches, which shares a similar trend with the correlation computations in Fig. 8(a). When $w = 1440$, the running time for approximate approach is 0.94s, which is an order of magnitude faster than 9.3s of the exact approach. Fig. 8(c) 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. Fig. 8(d) also presents the approximation error rate over time when we move forward the sliding window by setting $w = 360$, $\gamma = 0.3$. It shows that the error is always lower than 0.15 as time goes far away from the initial time tick. This justifies our approximate approach refines peak values and can achieve good approximation accuracy.

6 Related Work

There are several existing studies on multiple time series stream mining. Spiros et al. [11] tracked local correlations by comparing the local auto-covariance matrices of each time series. Zhu and Shasha [18] monitored thousands of time series data but focused on finding high cross-correlation pairs of them. Tan et al. [14] analyzed the linear correlation of multiple climate time series and attemptedX to construct climate index using clustering. Sakurai et al. [13] 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 [6, 17, 8, 9] that constructs a weighted graph on time series in order to discover different interesting patterns. Dorr and Denton [6] proposed to construct a hierarchic graph by analyzing similar subsequence of time series to discover timing patterns (e.g., a subsequence of one time series ”begins earlier”, ”ends later”, or is ”longer” than another). Idé and Kashima [8] proposed an anomaly detection method by analyzing the eigenspace of the dependency matrix. Later, Idé et al. [9] computed the anomaly score of a time series by investigating its k-neighborhood time series. Instead, our work discovers leaders by constructing a graph based on the lead-lag relations of time series.

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 climate 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.

Acknowledgment: The work was supported by grants of the Research Grants Council of the Hong Kong SAR, China No. 419008 and 419109.

References

1. R. Bhuyan. Information, alternative markets, and security price processes: A survey of literature. Finance 0211002, EconWPA, 2002.
2. G. Box, G. M. Jenkins, and G. Reinsel. *Time Series Analysis: Forecasting and Control*. Prentice Hall, 1994.
3. R. P. Brent. *Algorithms for Minimization Without Derivatives*. Dover Publications, 2002.
4. S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Comput. Netw. ISDN Syst.*, 30(1–7):107–117, 1998.
5. K. Chan. A further analysis of the lead-lag relationship between the cash market and stock index futures market. *Review of Financial Studies*, 5(1):123–152, 1992.
6. D. H. Dorr and A. M. Denton. Establishing relationships among patterns in stock market data. *Data & Knowledge Engineering*, 2008.
7. C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–38, July 1969.
8. T. Idé and H. Kashima. Eigenspace-based anomaly detection in computer systems. In *KDD*, pages 440–449, 2004.
9. T. Idé, S. Papadimitriou, and M. Vlachos. Computing correlation anomaly scores using stochastic nearest neighbors. In *ICDM*, pages 523–528, 2007.
10. E. Meijering. Chronology of interpolation: From ancient astronomy to modern signal and image processing. In *Proc. of the IEEE*, pages 319–342, 2002.
11. S. Papadimitriou, J. Sun, and P. S. Yu. Local correlation tracking in time series. In *ICDM*, pages 456–465, 2006.
12. P. Säfvenblad. Lead-lag effects when prices reveal cross-security information. Working Paper Series in Economics and Finance 189, Stockholm School of Economics, Sept. 1997.
13. Y. Sakurai, S. Papadimitriou, and C. Faloutsos. Braid: Stream mining through group lag correlations. In *SIGMOD*, pages 599–610, 2005.
14. M. Steinbach, P.-N. Tan, V. Kumar, S. A. Klooster, and C. Potter. Discovery of climate indices using clustering. In *KDD*, pages 446–455, 2003.
15. P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2006.
16. H. von Storch and F. W. Zwiers. *Statistical Analysis in Climate Research*. Cambridge University Press, 2002.
17. J. D. Wichard, C. Merkwirth, and M. Ogorzałlek. Detecting correlation in stock market. *Physica A: Statistical Mechanics and its Applications*, 344(1-2):308–311, 2004.
18. Y. Zhu and D. Shasha. Statstream: Statistical monitoring of thousands of data streams in real time. In *VLDB*, pages 358–369, 2002.