Lag correlation analysis based on Boolean presentation over multiple data streams

Dejun Yue Tiancheng Zhang Ge Yu Yu Gu

School of Information Science and Engineering, Northeastern University, ShenYang 110004, P. R. China

Abstract

Correlation analysis is a basic problem in the field of data stream mining. In this paper we propose a new method based on Boolean representation for lag correlation analysis among multiple data streams. The raw stream sequence is transformed into the Boolean sequence, and the lags in any correlation pairs of sequences can be easily gained by simple bit operations. Compared with pair-wise approach, this proposed method can get the exact result more efficiently by reducing huge calculation in very limited space. Both the theory analysis and the experimental evaluations show that this method has great computation complexity with high accuracy.

Keywords: Lag correlation, Boolean representation, data streams

1. Introduciton

Data streams have received considerable attention in various communities due to several important applications, such as network analysis, sensor network monitoring, financial data analysis, and scientific data processing. In all these situations, huge amount of data arrive at high rates, which makes traditional database systems prohibitively slow.

Many recent efforts concentrate on summarization and pattern discovery in data streams[1][2][3][4][6][8][9]. Here we focus on fast lag correlation analysis among multiple data streams. In practice lag correlation are frequent and ubiquitous, for example, a decrease of a certain stock price typically precedes increases of some others by a few minutes. Lots of data streams in many applications are correlated with an unknown lag. Our goal is to monitor thousands of data streams and determine all pairs of streams that have lag correlations. Furthermore, we want to give the lag for each correlation pair effectively and efficiently.

Correlation analysis over a large number of streams is a challenging task because the data stream is always burst and endless. Lots of work has been done about how to monitor thousands of data streams[5][6][7][8]. But all of these methods are based on complex transformation which transforms the raw stream sequence into simple summarization, then computes the correlation by pair-wise way which leads to tremendous calculation costs because most pairs of sequences may have no correlations at all.

We propose a novel approach based on Boolean cross-correlation (**BCC**) method to discover the lag correlations among multiple data streams. The raw stream sequence is firstly transformed into the Boolean sequence which is just a bit sequence, and the correlation results can be easily gained by simple bit operations. With huge amount of data streams, this method can quickly allocate the correlation pairs of sequences as well as the lags in an efficient way by reducing huge calculation in a little space. We can demonstrate the correctness of this approach by sufficient theory analysis. The experimental evaluations show that our method has great computation performance with high accuracy

The rest of the paper is organized as follows: Section 2 introduces necessary definitions and notations. Section 3 presents our proposed method for lag correlation analysis. Section 4 gives our theoretical analysis for this method. Section 5 reviews the results of the experiments, which clearly show the effectiveness of this approach. Section 6 is a brief conclusion.

2. Preliminaries

Data streams can be regarded as continuous, infinite sequences. In practice, we just compare the recent values by sliding window model. Let X *be* a stream sequence $\{x_1,\ldots,x_t,\ldots,x_n\}$, where x_n is the most recent value, and the window's length is n. As time goes by, the values in the sliding window will be updated continuously. Our goal is to monitor m numerical sequences, $X_1, X_2,\ldots,X_m$, and to determine all the pairs of sequences that have a lag correlation and report the lag value.

We will adopt Pearson formula as the criteria for the lag correlation.

Definition 1 (*CCF*).Given two stream sequences X and Y are $\{x_1,\ldots,x_t,\ldots,x_n\}$ and $\{y_1,\ldots,y_t,\ldots,y_n\}$ respectively, then the cross-correlation function (*CCF*) of these two sequences is:

$$\rho(k)=\frac{\sum_{t=k+1}^{n}(x_{t-k}-\overline{x})(y_t-\overline{y})}{\sqrt{\sum_{t=1}^{n-k}(x_t-\overline{x})^2}\sqrt{\sum_{t=k+1}^{n}(y_t-\overline{y})^2}} \qquad (1)$$

$$\overline{x}=\frac{1}{n-k}\sum_{t=1}^{n-k}x_t,\ \overline{y}=\frac{1}{n-k}\sum_{t=k+1}^{n}y_t$$

Where $\rho(k)$ denotes the correlation coefficient, when Y is delayed by k. The symmetric case when Y is delayed can be handled the same. We restrict the maximum lag k to be $n/2$.

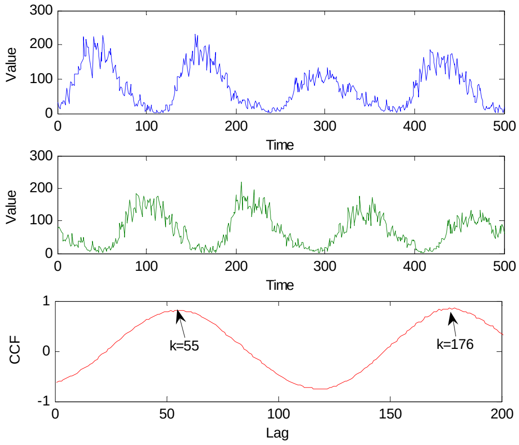


Fig 1: Example of lag correlation.

Definition 2 (Lag correlation). Two sequences X and Y have a lag correlation of k, if $\rho(k)$ is above a threshold ε, and is actually the earliest local maximum which is not multiples of any others.

From Fig 1 we can see that two sequences have a lag correlation when k=55, the *CCF* curve may have multiple local maximum for periodical reasons, such as k=176, clearly, the earliest lag the most important one.

Symbol	Definition
n	Length of sequence
m	Number of sequences
k	*Lag*
ρ	Correlation coefficient
δ	Boolean correlation coefficient
ε	Correlation threshold
ξ	Boolean correlation threshold
x_t	Value of a sequence X at time t=1,…, n

Table 1: Symbols and Definitions.

Naïve method to compute the *CCF* has huge calculation cost, there will be $O(n^2)$ time to just compare two sequences, what is worse, the time complexity will jump to $O(m^2n^2)$ for m sequences. In stream environment, it's unrealistic for this pair-wise method to analyze the correlations. In next section, we will propose a novel Boolean cross-correlation method to solve this problem efficiently.

3. Boolean cross-correlation

Boolean sequences can also be considered as a binary number. Each value in a Boolean sequence can only be 0 or 1. Boolean representation occupies less storage memory because one byte in computer contains only 8 bits, Boolean representation provides excellent time and space complexity for the data streams processing.

3.1. Boolean representation

For the purpose of accommodating the feature of data streams, we give a novel approach which transforms the raw sequence into a Boolean sequence based on [4]. Our proposed representation works by replacing each real valued data with a single bit, which means each value in sequences can only be 0 or 1.

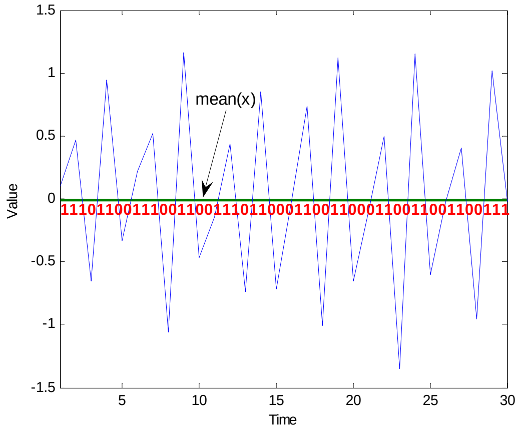


Fig 2: Boolean representation.

Definition 3 (Boolean sequence). Let X: $\{x_1,\ldots,x_t,\ldots,x_n\}$ be a stream sequence with the length of n, then the corresponding Boolean sequence W is $\{w_1,\ldots,w_t,\ldots,w_n\}$, where

$$w_t=\begin{cases}1, x_t>\overline{x}\\0, x_t\le\overline{x}\end{cases}\qquad \overline{x}=\frac{1}{n}\sum_{t=1}^{n}x_t \qquad (2)$$

From Fig 2 we know that the Boolean sequences can be easily gained by comparison operation with the mean values of the raw sequences. We can intuitively think that the Boolean sequences can reflect the main trends. Next, we'll give the definition of Boolean cross-correlation function based on Boolean representation.

Definition 4 (*BCCF*).Given two Boolean sequences W and V are $\{w_1,\ldots,w_t,\ldots,w_n\}$ and $\{v_1,\ldots,v_t,\ldots,v_n\}$ respectively, then the Boolean cross-correlation function (*BCCF*) of these two sequences is:

$$\delta(k) = 1 - \frac{\sum_{t=k+1}^{n} w_{t-k} \oplus v_t}{n-k} \tag{3}$$

where $\oplus$ is ***XOR*** operation, $\delta(k)$ denotes the correlation coefficient, when V is delayed by k.

Definition 5 (Lag correlation). Given two Boolean sequences W and V, $\delta(k)$ is their Boolean correlation coefficient and ξ is a Boolean correlation threshold, then W and V have a lag correlation of k, if $\delta(k)>\xi$(or $\delta(k)<1-\xi$) and $\delta(k)$ is actually the earliest local maximum which is not multiples of any others, where $\delta(k)<1-\xi$ denotes the negative correlation.

Without complex pair-wise comparison of raw streams sequences, we just need to operate on the corresponding Boolean sequences with simple Boolean computations by equation (3).

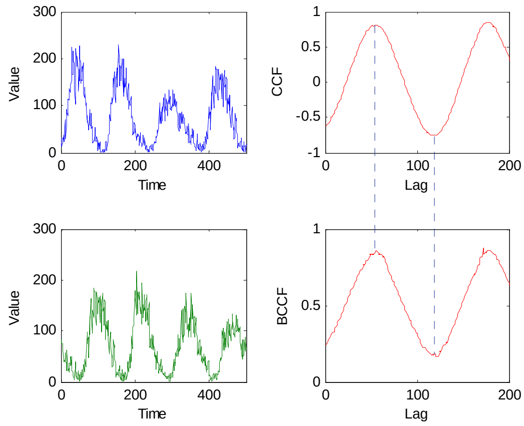


Fig.3: Comparison of *BCCF* and *CCF*.

From Fig3, we can see that *BCCF* curve has almost the same shape with the *CCF* curve. The lag time detected by *BCCF* is approximate the according point discovered by *CCF*. In next section, we will give a theoretical demonstration that *BCCF* can reflect almost the same trends as the *CCF*, we can simply get lag correlation results just by simple *BCCF*.

3.2. The algorithm of BCC

```
Algorithm 1: BooleanTrans /*Transform the raw sequence into Boolean sequence*/
    input: raw sequence X: ($x_1$,..., $x_t$ ,..., $x_n$), mean value : $\bar{x}$ ;
    output: Boolean sequence W:($w_1$,..., $w_t$ ,...,$w_n$).
      for t=1 to n do
        { if($x_t > \bar{x}$ )
            $w_t$=1
          else
            $w_t$=0}
          return W}
    Algorithm2: BCC
    input: stream sequences set D ($X_1$, $X_2$, ..., $X_M$), $\xi$ (Boolean correlation threshold);
    output: correlation set C (correlation pairs and the lags).
      for (each sequence $X_i$ in D) do
      {  compute the mean value $\bar{x}_i$ ;
            $W_i$=BooleanTrans ($X_i$ , $\bar{x}_i$ );
            add $W_i$ into the Boolean sequence set B; }
       for (for each pair ($W_i$ , $W_j$) in B)do
        {  for k=1 to n/2 do
             {$\delta_k$=ComputeBCCF ($W_i$ , $W_j$ ,k); }
               /*compute the Boolean cross-correlation function*/
       for k=1 to n/2 do
         { if (($\delta_k$>ξ or $\delta_k$<1-ξ) and
              (($\delta_k$<$\delta_{k-1}$ and $\delta_k$<$\delta_{k+1}$) or
              ($\delta_k$>$\delta_{k-1}$ and $\delta_k$>$\delta_{k+1}$)))
         /* the local extremum */
       add ($X_i$ ,$X_j$ ,k) into C; }
       output C
```

4. Theoretical analysis

4.1. Precision

In this section we give a theoretical analysis to show the accuracy of the **BCC** method. Again, we focus on two sequences X and Y. To simplify the discussion without loss of generality, we assume the given sequences have normal distribution.

Lemma1. Let$\{X_t\}$,$\{Y_t\}$ be both standard normal distribution sequences, we have $E(X_t)=E(Y_t)=0$, $D(X_t)= D(Y_t)=1$, $\{W_t\}$ and $\{V_t\}$ are Boolean sequences of $\{X_t\}$ and $\{Y_t\}$ respectively, assume $\varphi_k=P(W_{t-k}=1|V_t=1)$, then the Pearson correlation coefficient of $\{W_t\}$ and $\{V_t\}$:

$$\rho_k= 2\varphi_k-1$$

Proof. From the symmetry of the standard normal distribution, we know that:
$P(W_t=1)=P(W_t=0)=0.5$, $E(W_t)=0.5$,
$D(W_t)= E(W_t^2)- [E(W_t)]^2=0.5-0.25=0.25$, so
$P(V_t=1)$, $E(V_t)=0.5$, $D(V_t)=0.25$,

$$E(W_{t-k}V_t)= P(V_t=1)P(W_{t-k}=1|\ V_t=1)= \frac{1}{2}\varphi_k$$

Then, we can get the covariance of $\{W_{t-k}\}$ and $\{V_t\}$:
$COV(W_{t-k},\ V_t)=E[(W_{t-k}-0.5)(V_t-0.5)]$
$=E(W_{t-k}\ V_t)-0.25=\frac{1}{2}\varphi_k-\frac{1}{4}$, thus
$\rho_k=COV(W_{t-k},V_t)\big/\sqrt{D(W_{t-k})D(V_t)}$ $=2\varphi_k-1$. □

Theorem 1. Let $\{X_t\}$,$\{Y_t\}$ be both standard normal distribution sequences, $E(X_t)=E(Y_t)=0$, $D(X_t)=D(Y_t)=1$， $\{W_t\}$ and $\{V_t\}$ are Boolean sequences of $\{X_t\}$ and $\{Y_t\}$, let ρ_k be the Pearson correlation coefficient of $\{X_t\}$ and $\{Y_t\}$, and $\rho_k^{W_tV_t}$ be the Pearson correlation coefficient of $\{W_t\}$and$\{V_t\}$, then:

$$\rho_k=\sin(\frac{\pi}{2}\rho_k^{W_tV_t}) \tag{4}$$

Proof. The joint probability density function of $(X_t, Y_t\}$ is:

$$f(x,y)=\frac{1}{2\pi\sqrt{1-\rho_k^2}}\exp\left\{-\frac{1}{2(1-\rho_k^2)}(x^2+y^2-2\rho_k xy)\right\}$$

$$E(W_{t-k}V_t)=P(W_{t-k}=1,V_t=1)=P(X_{t-k}>0,Y_t>0)$$

$$=\frac{1}{2\pi\sqrt{1-\rho_k^2}}\int_0^\infty\int_0^\infty\exp\left\{-\frac{1}{2(1-\rho_k^2)}(x^2+y^2-2\rho_k xy)\right\}dxdy$$

$$=\frac{1}{2\pi\sqrt{1-\rho_k^2}}\int_0^\infty\int_0^{\frac{\pi}{2}}r\exp\left\{-\frac{1}{2(1-\rho_k^2)}(1-\rho_k\sin 2\theta)\right\}d\theta dr$$

$$=\frac{1}{2\pi\sqrt{1-\rho_k^2}}\int_0^{\frac{\pi}{2}}\frac{1}{1-\rho_k\sin 2\theta}d\theta=\frac{1}{4}+\frac{1}{2\pi}\arcsin\rho_k$$

From lemma 1, we can get:

$$\rho_k^{W_tV_t}=2\varphi-1=4E(W_{t-k}V_t)-1=\frac{2}{\pi}\arcsin\rho_k$$

$$\rho_k=\sin(\frac{\pi}{2}\rho_k^{W_tV_t})$$ □

From theorem1, we know that $\rho_k^{W_tV_t}$ has the same monotony interval with ρ_k, and ρ_k has a linear relation with φ_k, we can get the lag correlation from the ρ_k as well as φ_k because they have the same trends. Next, we will give a novel method to calculate the approximate value of ρ_k.

Theorem 2. Let $\{X_t\}$,$\{Y_t\}$(t=1,2,…n) be finite length standard normal distribution sequences, $\{W_t\}$ and$\{V_t\}$ are Boolean sequences of $\{X_t\}$ and $\{Y_t\}$ respectively, $\varphi_k=P(W_{t-k}=1|\ V_t=1)$, k denotes the lag value, at an arbitrary time t, let $W_t=w_t,\ V_t=v_t$, then the maximum likelihood of φ_k:

$$\hat{\varphi}_k=\frac{n-k-\sum_{t=k+1}^{n}(v_t+w_{t-k})+2\sum_{t=k+1}^{n}v_tw_{t-k}}{n-k} \tag{5}$$

Proof. From the given condition, we have

$$P(W_1=w_1,\dots,W_{k+1}=w_{k+1};V_1=v_1,\dots,V_{k+1}=v_{k+1})$$
$$P(W_2=w_2,\dots,W_{k+2}=w_{k+2};V_2=v_2,\dots,V_{k+2}=v_{k+2})\dots$$
$$P(W_{n-k}=w_{n-k},\dots,W_n=w_n;V_{n-k}=v_{n-k},\dots,V_n=v_n)$$
$$=P(W_1=w_1)P(V_{k+1}=v_{k+1}\,|\,W_1=w_1)P(W_2=w_2,\dots,$$
$$V_k=v_k\,|\,W_1=w_1,V_{k+1}=v_{k+1})\dots P(W_{n-k}=w_{n-k})$$
$$P(V_n=v_n\,|\,W_{n-k}=w_{n-k})P(w_{n-k+1}=w_{n-k+1},\dots,$$
$$V_{n-1}=v_{n-1}\,|\,W_{n-k}=w_{n-k},V_n=v_n)$$
$$=\prod_{t=k+1}^{n}P(W_{i-k}=w_{i-k})\prod_{t=k+1}^{n}P(V_i=v_i\,|\,W_{i-k}=w_{i-k})$$
$$\cdot\prod_{t=k+1}^{n}P(W_{i-k+1}=w_{i-k+1},\dots,V_{i-1}=v_{i-1}\,|\,W_{i-k}=w_{i-k},V_i=v_i)$$

since $\varphi_k=P(W_{t-k}=1|V_t=1)=P(W_{t-k}=0|V_t=0)$,
$1-\varphi_k=P(W_{t-k}=1|\ V_t=0)=P(W_{t-k}=0|\ V_t=1)$,

Then, the likelihood function of φ:

$$L(\varphi_k)=(\frac{1}{2})^{n-k}\varphi_k^{\,n-k-\sum_{t=k+1}^{n}(v_t+w_{t-k})+2\sum_{t=k+1}^{n}v_tw_{t-k}}(1-\varphi_k)^{\sum_{t=k+1}^{n}(v_t+w_{t-k})-2\sum_{t=k+1}^{n}v_tw_{t-k}}$$
$$\cdot\prod_{t=k+1}^{n}P(W_{i-k+1}=w_{i-k+1},\dots,V_{i-1}=v_{i-1}\,|\,W_{i-k}=w_{i-k},V_i=v_i)$$

Let $\frac{\partial\ln L(\varphi_k)}{\partial\varphi_k}=0$, we can get the maximum likelihood of φ_k:

$$\hat{\varphi}_k=\frac{n-k-\sum_{t=k+1}^{n}(v_t+w_{t-k})+2\sum_{t=k+1}^{n}v_tw_{t-k}}{n-k}$$ □

Each item in Boolean sequence is just a bit. We can replace the algebra calculation with simple bit operation, to improve the effectiveness of the calculation. Equation (5) can be transformed as follows:

$$\frac{n-k-\sum_{t=k+1}^{n}(v_t+w_{t-k})+2\sum_{t=k+1}^{n}v_tw_{t-k}}{n-k}=1-\frac{\sum_{t=k+1}^{n}v_t\oplus w_{t-k}}{n-k}$$

We can see that $\hat{\varphi}_k$ is just the Boolean cross-correlation coefficient $\delta(k)$.

From theorem 1 and theorem 2, we can get:

$$\hat{\rho}_k^{W_tV_t}=2\hat{\varphi}_k-1,\hat{\rho}_k=\sin(\frac{\pi}{2}\hat{\rho}_k^{W_tV_t})=\sin\pi(\hat{\varphi}_k-\frac{1}{2})$$

thus, $$\hat{\rho}_k=\sin\pi(\delta_k-\frac{1}{2}) \tag{6}$$

Above all, cross-correlation function has the same trend with Boolean cross-correlation coefficient with infinite length sequences. We need not compute the concrete value of *CCF*, but get the lag correlation result by quickly allocating the earliest maximum in *BCCF* curve.

4.2. Complexity

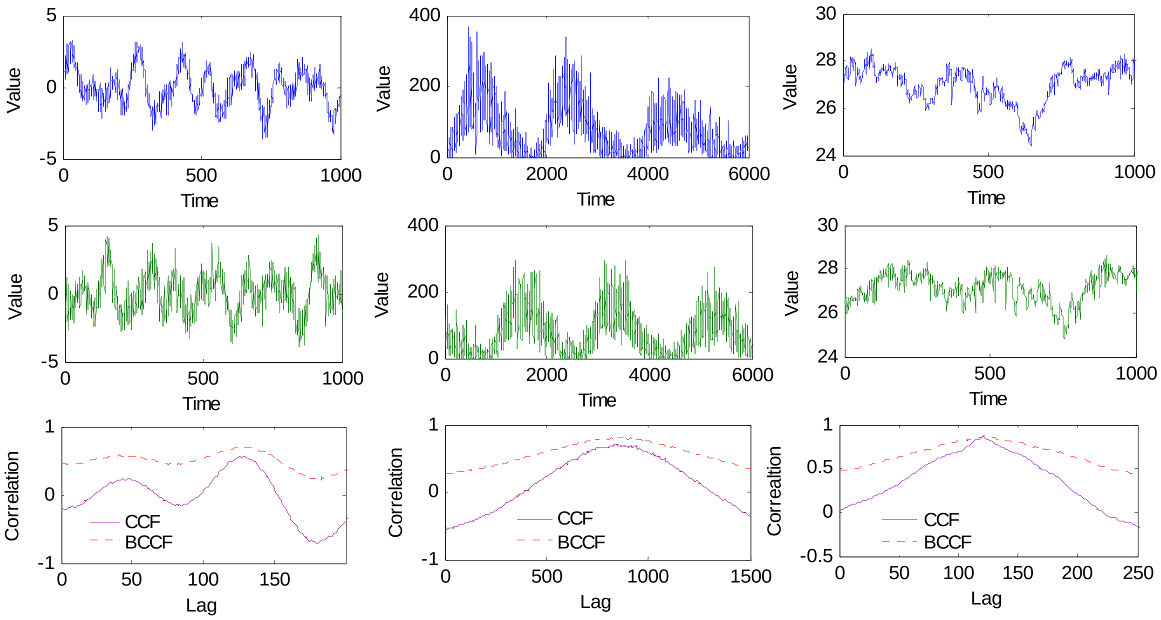
Boolean sequences can be considered as a binary number which occupies less storage memory because one byte in computer contains only 8 bits, from [4] we know that this method can yield compression ratios from 32:1 to better than 1000:1.So we can get unbelievable space complexity which is far below $O(logn)$. And above all, each item in Boolean sequences is just a bit and bit operations are very efficient and effective for data streams. We just need to replace the real value algebraic calculations with simple bit operations to reduce time complexity.

The main cost of **BCC** is the transformation to Boolean sequences. The procedure from real value to bits just require $O(mn)$ time for m sequences with each length of n. Then we need $(m\text{-}1)^2n/2$ times ***XOR*** operation between binary sequences, and the time requirement for efficient bit operations can be neglected. Moreover, we can also speed up the operation to sum the bits. Any algorithm to count the bits is $O(n)$. However, we can improve the constant terms in the time complexity function by using shift operators to evaluate value of each eight or sixteen bit sequence, then using a lookup table to find the number. This mechanism makes the calculation approximately five to ten times faster than ordinary counting.

Braid[6] need $O(1)$ time to analyze the lag correlation between two sequences, but it also need to compute the results by pair-wise way among multiple streams which leads to tremendous calculation costs because most pairs of sequences may have no correlations at all. But the pair-wise way for **BCC** is just simple bit operations which quickly allocate the correlation pairs effectively with high accuracy.

5. Experiments evaluation

To evaluate the effectiveness of **BCC** method, we performed experiments on real and synthetic datasets on a 2.4GHz Pentium 4 PC with 512 MB of main memory.



(a) Synthetic

(b) Sunspots

(c) TAO

Fig 4: Comparison of *BCCF* and *CCF*.

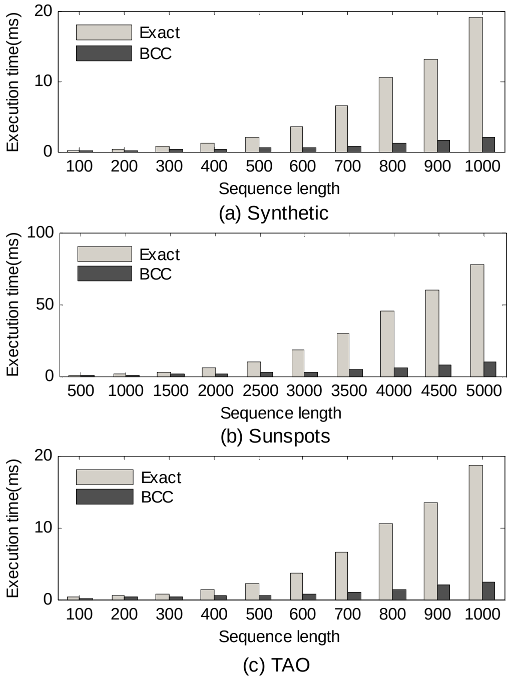
5.1. Precision

We compared **BCC** method with the implementation of naïve *CCF*. The datasets used are the following:

- *Synthetic*: the data set consists of two sequences of length n=1000, each sequence is a mixture of sine waves of different frequencies.
- *Sunspots*: number of sunspots per day. We choose two intervals form the dataset, each length n=6000.
- *TAO*: this dataset (Tropical Atmosphere Ocean) contains the air temperature of two sites in pacific from 2004 to 2006, obtained from the Pacific Marine Environmental Laboratory. (http://www.pmal.noaa.gov/tao)

Fig 4 shows the estimation of our proposed method for all data sets. In these figs, '*CCF*', '*BCCF*' denote the cross-correlation function curve and

Boolean cross-correlation function curve respectively. We can see that *BCCF* curve perfectly approximates the main trends of *CCF* curve for all the datasets. We can easily capture the local maximum in the *BCCF* curve which perfectly corresponds to lag correlation points shown in *CCF*.



(a) Synthetic

(b) Sunspots

(c) TAO

Fig.5:.Execution time of **BCC** and exact method with different sequence length

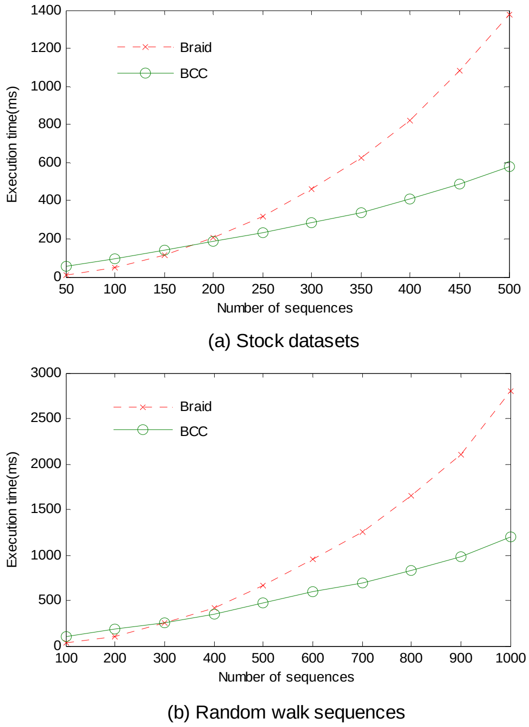
Table 2 shows the precision of the lag correlations captured by *BCCF* in all the datasets. We assume the lag correlations captured by *CCF* is the exact value and the precision will be obtained by making a comparison. The results are so perfect that all the values are above 98%.

Datasets	Lag correlation		Precision (%)
	CCF	BCCF	
Synthetic	127	128	99.2
Sunspots	855	864	98.9
TAO	119	119	100

Table 2:.Precision of BCC.

5.2. Performance

We theoretically discussed the complexity of **BCC** is Section 4.2, we did an empirical study of the computation time. First, we used the datasets in Section 5.1 to compare the execution time of **BCC** with exact pair-wise method. For arbitrary two sequences, **BCC** need $O(n)$ time compared with $O(n^2)$ for naive method. As shown in Fig 5, the execution time of exact method increases in a quadratic way as the sequence length continues to grow. However, instead of the $O(n^2)$ that the naïve implementation requires, **BCC** keep an approximate linear trend which achieve a dramatic reduction in computation time.



(a) Stock datasets

(b) Random walk sequences

Fig.6:.Execution of **BCC** and Braid with different number of sequences

We also compared **BCC** with Braid[6] with the same experimental environment. The datasets contain two parts: 1000 sequences generated by random walk model and financial sequences consist of the open prices of 500 stocks in China from 2002 to 2006, with each length n= 2500.

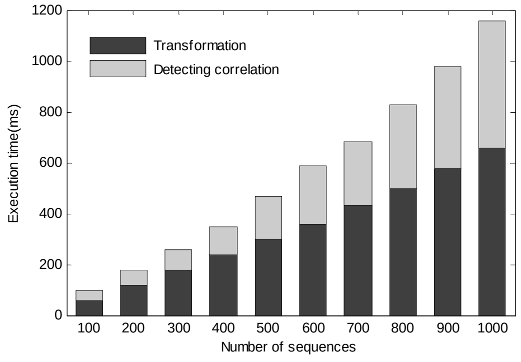


Fig.7: Execution time of **BCC** for random walk sequence

From Fig 6, we can see that Braid has great computation time when the number of sequences is small because it just need $O(1)$ time. However, as the sequence number grows up, the execution time of Braid has a quadratic increase for pair-wise way. Meanwhile, **BCC** also keep a linear time because the main costs come from the transformation to the Boolean sequence shown in Fig 7, and the pair-wise comparison cost is just simple bit operations which can be neglected.

6. Conclusion

In this paper, we propose a novel approach based on Boolean cross-correlation (**BCC**) method to discover the lag correlations of multi-stream time sequences. Each raw stream sequence is firstly transformed into the Boolean sequence which is just a binary number, and the lags of any correlation pairs can be easily gained by simple bit operations. Compared with traditional pair-wise approach, this method can get the quickly allocate the exact result more efficiently by reducing huge calculation in very limited space especially when the number of streams is very large. Both the theory analysis and the experimental evaluations show that this algorithm has great computation complexity with high accuracy.

Acknowledgement

This work is partially supported by Program for New Century Excellent Talents in University (NCET), National Nature Science Foundation of China (Grant No. 60503036, No. 60473073, No. 60573090 and No. 60673139) and Fok Ying Tung Education Foundation under Grant No. 104027.

References

[1] V. Megalooikonomou, G. Li. , Wang, and Q, A dimensionality reduction technique for efficient similarity analysis of time series databases. In *Proceeding of the 13th ACM CIKM international conference on information and knowledge management*, Washington, DC, Nov.8-13, pp.160-161, 2004.

[2] M. –J. Hsieh, M. –S. Chen, and P. S. Yu., Integrating DCT and DWT for approximating cube streams. In *CIKM*, 2005.

[3] S. Guha, C. Kim, and K. Shim, Xwave: Approximate extended wavelets for streaming data. In *VLDB*, pp.288-299, 2004.

[4] A. Ratanamahatana, E. Keogh, A. Bagnall and S. Lonardi, A novel bit level time series representation with implication for similarity search and clustering. In *Proceeding 9th Pacific-Asian Int. Conf. on Knowledge Discovery and Data Mining (PAKDD'05)*, Hanoi, Vietnam, 2005.

[5] Y. Zhu and D. Shasha, StatStream: Statistical monitoring of thousands of data streams in real time. In *Proc. of the 28th Int'l Conf. on Very Large Data Bases*, Hong Kong: Morgan Kaufmann, pp.358-369 ,2002

[6] Y. Sakurai, S.Panadimitriou, J. Sun and C. Faloutsos, Braid: Stream mining through group lag correlations. In *Proceedings of ACM SIGMOD*, Baltimore, Maryland, pp.599-610, 2005.

[7] S. Guha, D.Gunopulos and N.Koudas, Correlating synchronous and asynchronous data streams. In *Proc. of the 9th Int'l Conf. on Knowledge Discovery and Data Mining*, Washington DC: ACM Press, pp.529-534, 2003.

[8] S. Papadimitriou, J. Sun and C, Faloutsos. Stream pattern discovery in multiple time series. In *Proceeding of the 31st VLDB Conference*, Trondheim, Norway, 2005.

[9] S. Papadimitriou and P. S. Yu, Optimal multi-scale patterns in time series streams. In *Proceeding of SIGMOD*, Chicago, Illinois, USA, 2006.