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. Introduction

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 sequences that have lag correlations. Furthermore, we want to give the lag for each correlation pair effectively and efficiently.

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, x_2, ..., x_n\} and \{y_1, y_2, ..., y_m\} respectively, then the cross-correlation function (CCF) of these two sequences is:
\[
\rho(k) = \frac{\sum_{i=1}^{n-k} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n-k} (x_i - \bar{x})^2 \sum_{i=1}^{n-k} (y_i - \bar{y})^2}}
\]

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 \( n/2 \).

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.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n )</td>
<td>Length of sequence</td>
</tr>
<tr>
<td>( m )</td>
<td>Number of sequences</td>
</tr>
<tr>
<td>( k )</td>
<td>Lag</td>
</tr>
<tr>
<td>( \rho )</td>
<td>Correlation coefficient</td>
</tr>
<tr>
<td>( \delta )</td>
<td>Boolean correlation coefficient</td>
</tr>
<tr>
<td>( \varepsilon )</td>
<td>Correlation threshold</td>
</tr>
<tr>
<td>( \zeta )</td>
<td>Boolean correlation threshold</td>
</tr>
<tr>
<td>( x_t )</td>
<td>Value of a sequence ( X ) at time ( t=1,\ldots,n )</td>
</tr>
</tbody>
</table>

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.

Fig 1: Example of lag correlation.

Fig 2: Boolean representation.

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

\[
w_i = \begin{cases} 1, x_i > \bar{x} \\ 0, x_i \leq \bar{x} \end{cases} \quad \bar{x} = \frac{1}{n} \sum_{t=1}^{n} x_t
\]

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 will give the definition of Boolean cross-correlation function based on Boolean representation.
Definition 4 (BCCF). Given two Boolean sequences \( W \) and \( V \) are \( \{w_1, \ldots, w_n\} \) and \( \{v_1, \ldots, v_n\} \) respectively, then the Boolean cross-correlation function (BCCF) of these two sequences is:
\[
\delta(k) = \sum_{t=1}^{n} w_{t-k} \oplus v_t \tag{3}
\]
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 \( \zeta \) is a Boolean correlation threshold, then \( W \) and \( V \) have a lag correlation of \( k \), if \( \delta(k) > \zeta \) (or \( \delta(k) < 1 - \zeta \)) and \( \delta(k) \) is actually the earliest local maximum which is not multiples of any others, where \( \delta(k) < 1 - \zeta \) 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).

![Fig3: Comparison of BCCF and CCF.](image)

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 \( X: (x_1, \ldots, x_n) \), mean value : \( \bar{X} \);

**output:** Boolean sequence \( W: (w_1, \ldots, w_n) \).

```
for t=1 to n do
    {  if(\( x_t > \bar{X} \))
        w_t=1
    else
        w_t=0
    return W
```

Algorithm 2: BCC

**input:** stream sequences set \( D (X_1, X_2, \ldots, X_n) \), \( \zeta \)

**output:** correlation set \( C \) (correlation pairs and the lags).

1. for (each sequence \( X_i \) in \( D \)) do
   2. \{ compute the mean value \( \bar{X}_i \); \}
   3. \( W_i=\text{BooleanTrans} (X_i, \bar{X}_i) \);
   4. add \( W_i \) into the Boolean sequence set \( B \); \}
   5. for (each pair \( (W_i, W_j) \) in \( B \)) do
      6. \{ for \( k=1 \) to \( n/2 \) do
          7. \{ \( \delta_{i,j}=\text{ComputeBCCF} (W_i, W_j, k) \); \}
            /*compute the Boolean cross-correlation function*/
          8. for \( k=1 \) to \( n/2 \) do
              9. \{ if (\( \delta_{i,j} > \zeta \) or \( \delta_{i,j} < 1 - \zeta \)) and
                  \((\delta_{i,j} < \delta_{k,j} \text{ and } \delta_{i,j} < \delta_{k,i}) \) or
                  \((\delta_{k,j} < \delta_{i,j} \text{ and } \delta_{k,i} > \delta_{i,j}))\)
                    /* the local extremum */
                10. 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.

**Lemma.** Let \( \{X_i\}, \{Y_i\} \) be both standard normal distribution sequences, we have \( E(X_i) = E(Y_i) = 0 \), \( D(X_i) = D(Y_i) = 1 \), \( \{W_i\} \) and \( \{V_i\} \) are Boolean sequences of \( \{X_i\} \) and \( \{Y_i\} \) respectively, assume \( \rho = P(W_{x,k} = 1 | V_{y} = 1) \), then the Pearson correlation coefficient of \( \{W_i\} \) and \( \{V_i\} \):

\[
\rho = 2\rho - 1
\]

**Proof.** From the symmetry of the standard normal distribution, we know that:

\[
P(W_{x,1} = 1 | P(W_y = 0) = 0.5, E(W_y) = 0.5, D(W_y) = 0.5, E(W_y)^2 = 0.5 - 0.25 = 0.25, \text{ and so}
P(V_{y,1} = 1), E(V_y) = 0.5, D(V_y) = 0.25, D(V_y)^2 = 0.25, \text{ and so}
E(W_{x,k} | V_{y}) = P(V_{y} = 1)P(W_{x,k} = 1 | V_{y} = 1) = \frac{1}{2} \rho^k
\]
Then, we can get the covariance of \{W_{tk}\} and \{V_t\}:  
\[
\text{COV}(W_{tk}, V_t) = E[(W_{tk}-0.5)(V_t-0.5)]
\]
\[
= E(W_{tk}V_t) - 0.25 = \frac{1}{2} \rho_k - \frac{1}{4}, \quad \text{thus}
\]
\[
\rho_k = \frac{\text{COV}(W_{tk}, V_t)}{\sqrt{D(W_{tk})D(V_t)}} = 2\rho_k - 1. \quad \Box
\]

**Theorem 1.** Let \{X_t\}, \{Y_t\} be both standard normal distribution sequences, \(E(X_t) = E(Y_t) = 0\), \(D(X_t) = D(Y_t) = 1\), \(X_t\) and \(Y_t\) are Boolean sequences of \{X_t\} and \{Y_t\}, let \(\rho_k\) be the Pearson correlation coefficient of \{X_t\} and \{Y_t\}, and \(\phi_k\) the Pearson correlation coefficient of \{W_{tk}\} and \{V_t\}, then:

\[
\rho_k = \sin \left( \frac{\pi}{2} \rho_k \phi_k \right)
\]

**Proof:** The joint probability density function of \((X_t, Y_t)\) is:

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

\[
E(W_{tk}V_t) = P(W_{tk} = 1|V_t = 1) = P(X_{tk} > 0, Y_t > 0)
\]

\[
= \frac{1}{2\pi\sqrt{1-\rho_k^2}} \int_0^\infty \int_0^\infty \exp \left\{ -\frac{1}{2(1-\rho_k^2)} (x^2 + y^2 - 2\rho_k xy) \right\} \, dx \, dy
\]

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

\[
= \frac{1}{2\pi\sqrt{1-\rho_k^2}} \int_0^\infty \frac{1}{1-\rho_k \sin 2\theta} \, d\theta = \frac{\pi}{4} \arcsin \rho_k.
\]

From lemma 1, we can get:

\[
\rho_k^{\phi_k} = 2\rho_k - 1 = 4E(W_{tk}V_t) - 1 = \frac{2}{\pi} \text{arcsin } \rho_k
\]

\[
\rho_k = \sin \left( \frac{\pi}{2} \rho_k \phi_k \right) \quad \Box
\]

From theorem 1, we know that \(\rho_k^{\phi_k}\) has the same monotony interval with \(\rho_k\), and \(\rho_k\) has a linear relation with \(\phi_k\), we can get the lag correlation from the \(\rho_k\) as well as \(\phi_k\) because they have the same trends. Next, we will give a novel method to calculate the approximate value of \(\rho_k\).

**Theorem 2.** Let \(\{X_t\}, \{Y_t\}\) be finite length standard normal distribution sequences, \(\{W_{tk}\}\) and \(\{V_t\}\) are Boolean sequences of \(\{X_t\}\) and \(\{Y_t\}\) respectively, \(\phi_k = P(W_{tk} = 1|V_t = 1)\), \(k\) denotes the lag value, at an arbitrary time \(t\), let \(W_{tk} = W_{tk}\), \(V_t = V_t\), then the maximum likelihood of \(\phi_k\):

\[
\hat{\phi}_k = \frac{n-k-\sum_{t=k}^n (V_t + W_{tk}) + 2 \sum_{t=k}^n V_t W_{tk}}{n-k}
\]

**Proof:** From the given condition, we have

\[
P(W_j = w_{1,\ldots, W_{k+j} = w_{k+1}}; V_t = v_1, \ldots, V_{k+j} = v_{k+1})
\]

\[
= P(W_{2} = w_{2,\ldots, W_{k+2} = w_{k+2}}; V_2 = v_2, \ldots, V_{k+2} = v_{k+2})\ldots
\]

\[
P(W_{n-k} = w_{n-k}, \ldots, W_n = w_n; V_n = v_n) = P(W_{t} = w_t; V_t = v_t) P(W_{t+1} = w_{t+1}, \ldots, V_{k} = v_k)\ldots P(W_{n-k} = w_{n-k})
\]

\[
= \prod_{t=1}^n P(W_{t} = w_t) \prod_{t=1}^n P(V_t = v_t)
\]

since \(\phi_k = P(W_{tk} = 1|V_t = 1) = P(W_{tk} = 0|V_t = 0), 1 - \phi_k = P(W_{tk} = 1|V_t = 0) = P(W_{tk} = 0|V_t = 1),
\]

Then, the likelihood function of \(\phi_k\):

\[
L(\phi_k) = \frac{1}{2^n} \phi_k^{\sum_{t=k}^n (V_t + W_{tk})} \left( 1 - \phi_k \right)^{\sum_{t=k}^n (1-V_t) W_{tk}}
\]

\[
\frac{\partial \ln L(\phi_k)}{\partial \phi_k} = 0 \quad \text{we can get the maximum likelihood of } \phi_k:
\]

\[
\hat{\phi}_k = \frac{n-k-\sum_{t=k}^n (V_t + W_{tk}) + 2 \sum_{t=k}^n V_t W_{tk}}{n-k}
\]

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:

\[
n-k-\sum_{t=k}^n (V_t + W_{tk}) + 2 \sum_{t=k}^n V_t W_{tk} = 1 - \sum_{t=k}^n V_t \oplus W_{tk}
\]

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

From theorem 1 and theorem 2, we can get:

\[
\hat{\rho}_k = 2\phi_k - 1, \quad \hat{\rho}_k = \sin \left( \frac{\pi}{2} \rho_k \phi_k \right) = \sin \pi (\hat{\phi}_k - \frac{1}{2})
\]

thus, \(\hat{\rho}_k = \sin \pi (\hat{\delta}_k - \frac{1}{2}) \quad (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 $O(\log n)$. 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 $(m-1)\frac{n}{2}$ 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.

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.pmel.noaa.gov/tao](http://www.pmel.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

![Fig 4: Comparison of BCCF and CCF.](image-url)
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.

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%.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Lag correlation</th>
<th>Precision (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CCF</td>
<td>BCCF</td>
</tr>
<tr>
<td>Synthetic</td>
<td>127</td>
<td>128</td>
</tr>
<tr>
<td>Sunspots</td>
<td>855</td>
<td>864</td>
</tr>
<tr>
<td>TAO</td>
<td>119</td>
<td>119</td>
</tr>
</tbody>
</table>

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 naive implementation requires, BCC keep an approximate linear trend which achieve a dramatic reduction in computation time.

![Graph showing execution time of BCC and exact method with different sequence length](image1)

![Graph showing execution time of BCC and exact method with different number of sequences](image2)

![Graph showing execution time of BCC for random walk sequence](image3)

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 \( n=2500 \).
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


