Knowl Inf Syst (2008) 15:31–54
DOI 10.1007/s10115-006-0051-5

Knowledge and Information Systems

REGULAR PAPER

Zhen He · X. Sean Wang · Byung Suk Lee · Alan C. H. Ling

Mining partial periodic correlations in time series

Received: 7 February 2005 / Revised: 11 July 2006 / Accepted: 9 September 2006 /
Published online: 5 December 2006

Abstract Recently, periodic pattern mining from time series data has been studied extensively. However, an interesting type of periodic pattern, called partial periodic (PP) correlation in this paper, has not been investigated. An example of PP correlation is that power consumption is high either on Monday or Tuesday but not on both days. In general, a PP correlation is a set of offsets within a particular period such that the data at these offsets are correlated with a certain user-desired strength. In the above example, the period is a week (7 days), and each day of the week is an offset of the period. PP correlations can provide insightful knowledge about the time series and can be used for predicting future values. This paper introduces an algorithm to mine time series for PP correlations based on the principal component analysis (PCA) method. Specifically, given a period, the algorithm maps the time series data to data points in a multidimensional space, where the dimensions correspond to the offsets within the period. A PP correlation is then equivalent to correlation of data when projected to a subset of the dimensions. The algorithm discovers, with one sequential scan of data, all those PP correlations (called minimum PP correlations) that are not unions of some other PP correlations. Experiments using both real and synthetic data sets show that the PCA-based algorithm is highly efficient and effective in finding the minimum PP correlations.

Keywords Time series mining · Periodic patterns · Principal component analysis · Correlations

Z. He (✉)
Department of Computer Science, La Trobe University, Bundoora, VIC 3086, Australia
E-mail: z.he@latrobe.edu.au

X. S. Wang · B. S. Lee · A. C. H. Ling
Department of Computer Science, University of Vermont, Burlington, VT 05405, USA
E-mail: {Sean.Wang, Byung.Lee, Alan.Ling}@uvm.edu

1 Introduction

Finding periodicity in time series data is both a challenging and an important problem in many real world applications. Examples of time series that contain periodic patterns are numerous, including the time series of stock price, power consumption, sales data, meteorological data (e.g., temperature, humidity), etc. Many techniques have been developed for searching for periodic patterns in large time series data sets [1, 2, 8, 12, 13, 16, 18, 21–24].

Recently, several works have concentrated on finding *partial periodic* patterns [1, 8, 12, 13], which specify the periodic behavior of the time series at some, but not all, points in time within a certain period. This type of patterns appears frequently in real world applications. For example, within a period of 1 week in a stock price time series, the price of a particular stock may be high every Wednesday and low every Friday but not show any regularity on the other days. The usefulness of this type of patterns has been explained in the literature [1, 8, 12, 13].

The existing work on partial period patterns does not include an interesting type we call *partial periodic (PP) correlations*. Given a period (e.g., week) spanning a fixed number of offsets (e.g., 7 days), a PP correlation is a subset of these offsets such that the *correlation* of data at these offsets exceeds a user-given strength threshold. This type of pattern is what we study in this paper.

The following three examples illustrate how PP correlations may be used in practice. The first two examples assume the period of 1 week and the offset of one day.

Example 1 (Stock market analysis) Suppose a PP correlation has been found in the ticker price time series of a particular stock and it shows that the price is either high on Monday and low on Friday or the converse is true. Then, a stock market analyst may investigate the underlying cause of the PP correlation and find out that a major share holder regularly sells a large quantity of shares on Friday (thus causing the price to fall) and buys them all back on Monday (causing the price to rise) or does the reverse. The analyst can then take advantage of the information to his or her own profit.

Example 2 (Business logistics) Suppose a PP correlation has been found in the time series of the daily number of customers served at a certain restaurant and it shows that the sum of the numbers on Friday, Saturday and Sunday is roughly a constant. This information may be useful for the restaurant manager to budget the number of waiters accordingly for Saturday and Sunday after knowing the number of customers on Friday. After knowing the numbers on Friday and Saturday, it can be used for budgeting Sunday.

The third example assumes the period of 24 h and the offset of 1 h, with data collected only during weekdays.

Example 3 (Intelligent web caching) Consider a time series of the hourly number of visits to a particular web page by the client (web browser) of the user named John. Suppose a PP correlation has been found in the time series and it shows that the number of visits is high during either 10 a.m.–11 a.m. or 5 p.m.–6 p.m., but not both, on every weekday. This information may be used by the web client to prefetch the page just before 5 p.m. if the number of visits has been low during 10 a.m.–11 a.m.

Our experiments show that PP correlations occur frequently in real data (as will be shown in Sect. 5.2.3). However, no techniques capable of finding such patterns have appeared in the literature. This paper is the first attempt to introduce such a technique.

In our proposed technique, numerical time series data are mapped to a set of data points in a multidimensional space, where the dimensionality is the period size, and each dimension corresponds to one offset within the period.[1] A *PP correlation* is then defined as a subset of the dimensions such that the data at these offsets are correlated with a certain user-desired strength. For each PP correlation found, we can use a *linear constraint* to express the condition satisfied (within an error threshold) by the data appearing at the corresponding offsets within the period. Below we show two simple but useful cases.

Example 4 We illustrate PP correlations by using two cases that involve two offsets within a period: (1) PP negative correlation and (2) PP positive correlation. Consider stock price time series data mapped to data points in a seven-dimensional space $d_1 \times d_2 \times \cdots \times d_7$, where $d_1, d_2, \ldots, d_7$ are mapped from Monday, Tuesday, ..., Sunday, respectively. An example of PP negative correlation is the one in Example 1, where either the stock price is high on Monday and low on Friday or the converse is true, within a period of 7 days. The linear constraint for the PP correlation can then be expressed as $a_1 d_1 + b_1 d_5 + c_1 = 0$, where a_1 and b_1 are positive numbers. Note that d_1 corresponds to Monday and d_5 to Friday. Figure 1 illustrates this case using data points in ten periods. Consider only the data points mapped to the two-dimensional space $d_1 \times d_5$ (bottom of Fig. 1a) out of the time series given in the top of Fig. 1a. As shown in Fig. 1b, a PP negative correlation is found in the seemingly random time series. An example of PP positive correlation is that the stock price is either high on both Tuesday and Thursday or low on both days. The linear constraint for this PP correlation can be expressed as $a_2 d_2 - b_2 d_4 + c_2 = 0$, where a_2 and b_2 are positive numbers. Figure 2 illustrates this case using the same time series.

The number of PP correlations found in a time series may be very large, and in the worst case is exponential to the size of the period. In most cases, the user would not want to find all possible PP correlations. We believe that only a small fraction of the existing PP correlations are of interest to the user. This belief is based on two observations. First, PP correlations with a smaller number of correlated dimensions are more useful than those with a larger number. For instance, a PP correlation between the two dimensions d_4 and d_5 is more useful than one among the five dimensions d_1, d_2, d_3, d_4, and d_5. Indeed, in order to predict the value for d_5, the user needs to know only d_4 when using the former correlation but needs to know all of d_1, d_2, d_3, and d_4 when using the latter.

The second observation is that a PP correlation with a larger number of dimensions may not be so informative—hence not so useful—as those with a smaller number especially if the larger one can be inferred from the smaller ones. For instance, assume we have the PP correlation between d_1 and d_2 (with the linear

[1] We use the terms "offset" and "dimension" interchangeably in the sequel.

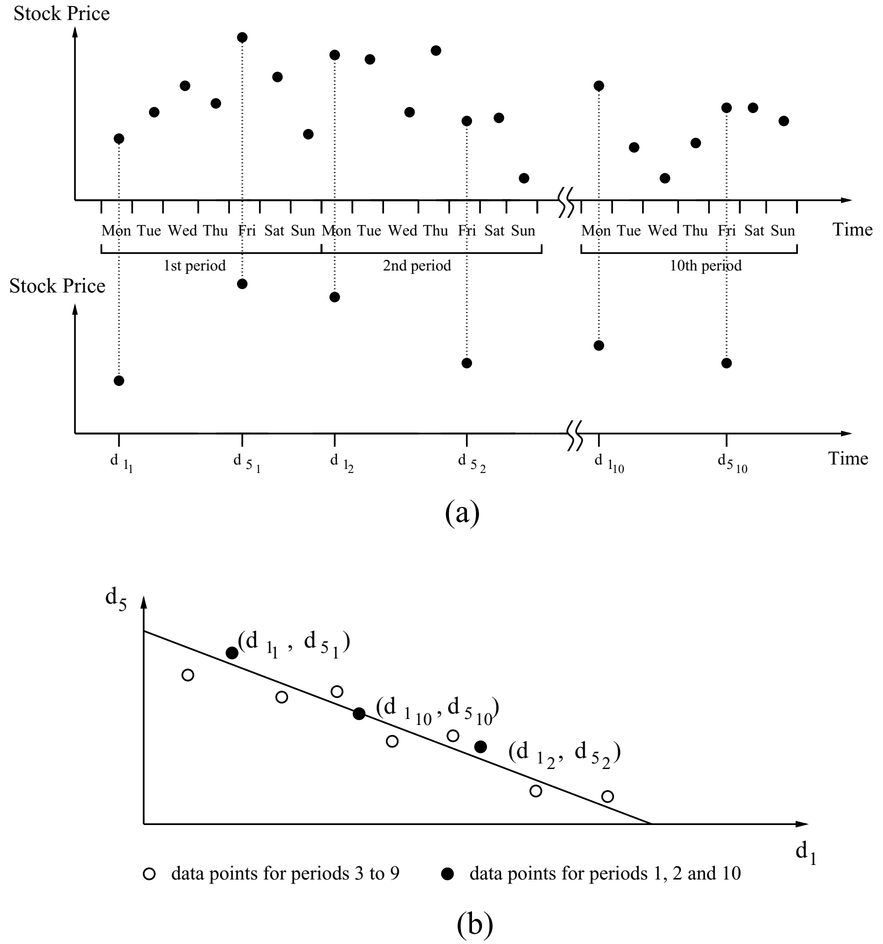


Fig. 1 An example of PP negative correlation

constraint $a_1d_1 + a_2d_2 = a_3$ where a_1, a_2, and a_3 are constants), and another between d_3 and d_4 (with the linear constraint $b_1d_3 + b_2d_4 = b_3$ where b_1, b_2, and b_3 are constants). Then we are likely to find a PP correlation among d_1, d_2, d_3, and d_4, with the likely linear constraint $a_1d_1 + a_2d_2 + b_1d_3 + b_2d_4 = a_3 + b_3$, although it is possible that the inferred PP correlation does not satisfy the user-desired strength even if the given two do.

Based on the above observations, we introduce the notion of *minimum PP correlations* in this paper. A PP correlation is said to be minimum if it is not the union of two or more other PP correlations. For example, a PP correlation with the dimensions $\{d_1, d_2, d_3\}$ is minimum if we do not have PP correlations among the subsets of $\{d_1, d_2, d_3\}$ such that the union of these subsets is $\{d_1, d_2, d_3\}$. On the contrary, the PP correlation is not minimum if we have PP correlations in the subsets $\{d_1, d_2\}$ and $\{d_2, d_3\}$

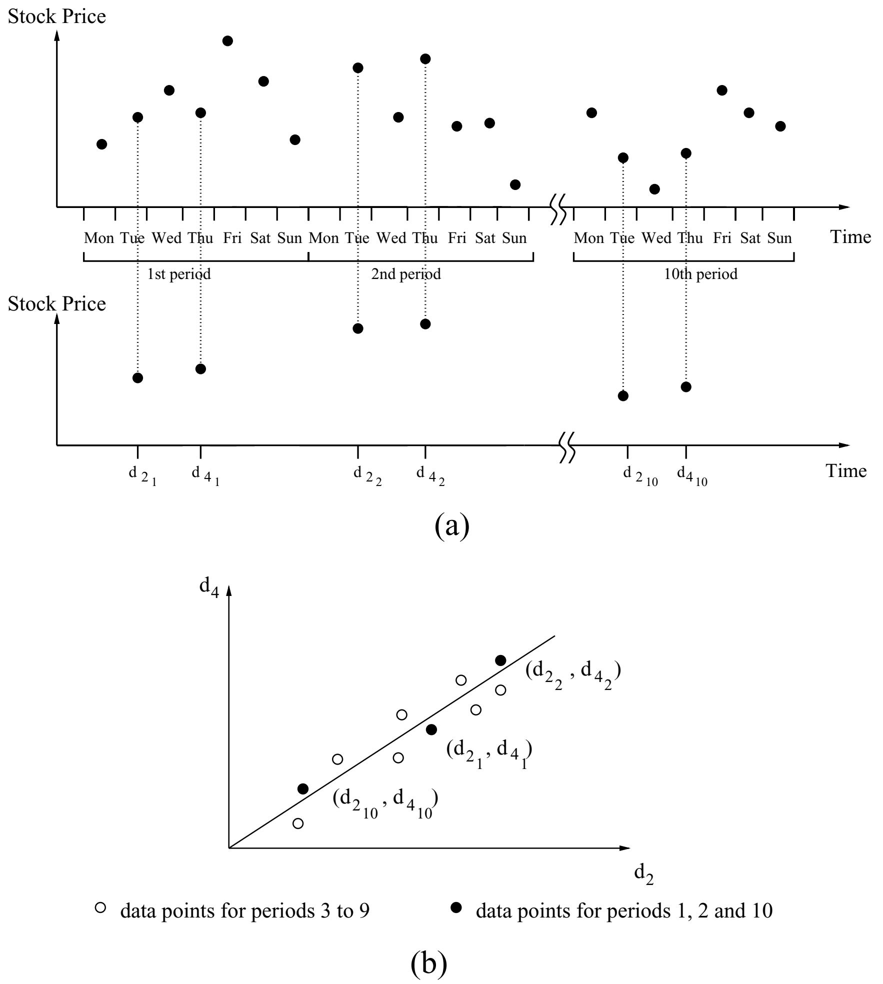


Fig. 2 An example of PP positive correlation

Mining for PP correlations involves two key issues: first, finding PP correlations for a given period size, and second, doing it in only one pass of a (large) time series. We address the first issue by exploiting a property of the *principal component analysis (PCA)*. More specifically, PCA can be used to find a vector (called *minimum variance PCA vector*) in a multidimensional space such that when the multidimensional points (the data) are projected onto the vector, the variance of the projections is minimized among all possible vectors in the space. Using this property, we can find the correlation with the *minimum variance*, in other words the correlation with the *maximum strength*[2]; If the variance of this correlation is below the variance threshold (hence, above the correlation strength threshold),

[2] We use the term "variance" in a technical discussion and "strength" in an intuitive discussion.

then a PP correlation is found with all the dimensions on which we have applied the PCA. The linear constraint for the found PP correlation is a hyperplane perpendicular to the minimum variance PCA vector.

The second issue is important due to the often prohibitive disk I/O cost of scanning a large time series multiple times. A naive application of PCA to mine PP correlations would require re-scanning the data for each subset of dimensions considered to apply the PCA. We address this issue by optimizing the PCA computation so that it takes only one pass through the time series to build a small amount of statistics (specifically, the covariance matrix) and use the statistics repeatedly to find all the minimum PP correlations without re-scanning the time series.

We have conducted experiments using both real and synthetic data sets. The results show that our one-pass approach leads to reduction of several orders of magnitude in the time to find all PP correlations, where the order increases with the increase of such parameters as the period size, data set size, and the maximum size of the PP correlations considered. We have also found that mining for *minimum* PP correlations produces significantly fewer PP correlations than mining for *all* PP correlations.

We make three contributions with this paper. First, we introduce the notion of PP correlations in time series. Second, we develop a PCA-based approach to finding all minimum PP correlations in one-pass scan of the time series. Third, we demonstrate the efficiency and efficacy of the PCA-based approach through experiments with both real and synthetic data sets.

We structure the rest of the paper as follows. Following this introduction, we discuss related work in Sect. 2, give an overview of the PCA and introduce key terms and concepts of PP correlations in Sect. 3, present our PCA-based approach to finding PP correlations in Sect. 4, and present the experiments in Sect. 5. Then, we present a simple extension of our PCA-based approach to consider multiple periods in Sect. 6, and conclude the paper with Sect. 7.

2 Related work

Two areas of related work are periodic pattern mining [1, 2, 8, 12, 13, 16, 18, 21, 22, 24] and forecasting [3, 4, 6, 9–11, 19]in time series.

As mentioned in the introduction, there has been significant work done in the area of mining periodic patterns in time series [1, 2, 8, 12, 13, 16, 18, 21–24]. Han et al. [12, 13] first introduced the notion of *partial periodic patterns* and presented two algorithms (in [12] and [13], respectively) for mining this type of patterns. Yang et al. [24] proposed mining *asynchronous* periodic patterns, which are patterns that may be present only within a subsequence and whose occurrences may be shifted due to “disturbance”. A number of papers [2, 8, 16, 18] center on discovering potential *periods* of time series data with high computational efficiency. Aref et al. [1] propose an *incremental* algorithm for mining partial periodic patterns. Yang et al. [22] mine “surprising periodic patterns” by using the concept of information gain to measure the overall degree of surprise within a data sequence. Wang et al. [21] propose an approach for mining patterns that are *hierarchical* in nature, where a higher level pattern consists of repeating lower level patterns. Jiong Yang et al. [23] propose efficient algorithms for mining high level patterns which would not be properly recognized by any existing method.

All existing works on finding periodic patterns (whether partial or total) assume discrete symbols in time series, not continuous numerical values as assumed in our work. Since symbols are not values that can be compared numerically, the methods of the existing works cannot find PP correlations as can be done with the technique of this paper.

There are several existing time series forecasting techniques, including general exponential smoothing [4], Holt's linear trends model [11], Holt–Winters seasonal model [6, 19], parametric regression [9, 10], Box–Jenkins [3], and linear models [3]. We focus our discussion on linear models here since they also use linear equations for characterizing the time series. There are many different linear models that are used for forecasting time series. They include autoregression [3], moving average [3], autoregression and moving average (ARMA) [3], integrated ARMA [3], and fractional integrated ARMA [3]. The basic ideas are illustrated by the ARMA(M,N) model, where the equation for value at time t, x_t, is given by

$$x_t = \sum_{m=1}^{M} a_m x_{t-m} + \sum_{n=0}^{N} b_n e_{t-n} \tag{1}$$

where M is the order of the autoregressive process, N is the order of the moving average process, $a_1, a_2, \ldots, a_M$ and $b_1, b_2, \ldots, b_N$s are constants used to fit the model, and $e_t, e_{t-1}, \ldots, e_{t-N}$ are the observable random shocks. In the ARMA and subsequent models, the linear models are designed solely for predicting the value at x_t based on a prior window of observations. In contrast, our work mines potentially useful knowledge from the time series by finding correlations that involve partial data (i.e., subsets of the offsets within a period) from the time series. The linear modeling techniques attempt to fit a single model (e.g., Eq. (1)) for forecasting *all* future values, whereas we mine a set of PP correlations that can be used for forecasting some future values at predictable points in time.

3 Preliminaries

In this section, we first give an overview of principal component analysis (PCA) techniques and then introduce important terms and concepts pertinent to PP correlations.

3.1 An overview of PCA

"Principal component analysis has been called one of the most valuable results from linear algebra" [20]. It is a simple non-parametric technique for extracting relevant information from a set of apparently random data. It is widely used for dimensionality reduction [5, 7, 15] and for finding correlations among attributes of data [17]. However, to our knowledge, PCA has never been used to look for PP correlations in time series data.

Given a set of k-dimensional data points, PCA finds a ranked set of orthogonal k-dimensional eigenvectors $\mathbf{v_1}, \mathbf{v_2}, \ldots \mathbf{v_i}, \ldots, \mathbf{v_k}$ (which we call *PCA vectors*) such that

- Each PCA vector is a unit vector, i.e., $\sqrt{{\beta_i}_1^2 + {\beta_i}_2^2 + \ldots + {\beta_i}_j^2 + \ldots + {\beta_i}_k^2} = 1$, where ${\beta_i}_j$ $(i, j = 1, 2, \ldots, k)$ is the jth component of the PCA vector $\mathbf{v_i}$.
- The variance along $\mathbf{v_i}$ is larger than or equal to the variance along $\mathbf{v_j}$ if $\mathbf{i} < \mathbf{j}$, where *variance along* a vector $\mathbf{v}$ is the variance of data points projected onto the vector $\mathbf{v}$.

The PCA vectors $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_k}$ can be computed by finding the linear transformation from the coordinates of the data points to new coordinates such that the covariance matrix of the new coordinates is diagonalized.

The linear transformation is given by the following equation:

$$\mathbf{VX} = \mathbf{Y} \tag{2}$$

where $\mathbf{V}$ is a matrix whose rows are the PCA vectors:

$$\mathbf{V} = \begin{bmatrix} \mathbf{v_1} \\ \mathbf{v_2} \\ \vdots \\ \mathbf{v_k} \end{bmatrix} \tag{3}$$

and $\mathbf{X}$ is a matrix whose columns are vectors of the coordinates of data points in the original k-dimensional space:

$$\mathbf{X} = \begin{bmatrix} X_{11} - \bar{X}_1 & \cdots & X_{1m} - \bar{X}_1 \\ X_{21} - \bar{X}_2 & \cdots & X_{2m} - \bar{X}_2 \\ \vdots & \ddots & \vdots \\ X_{k1} - \bar{X}_k & \cdots & X_{km} - \bar{X}_k \end{bmatrix} \tag{4}$$

where X_{ij} $(i = 1, 2, \ldots, k;\ j = 1, 2, \ldots, m)$ is the ith dimensional coordinate of the jth data point, $\bar{X}_i$ is the average ith dimensional coordinate (in order to give data points zero mean), and $\mathbf{Y}$ is a matrix whose columns are vectors of the new coordinates of the data points in the transformed k-dimensional space:

$$\mathbf{Y} = \begin{bmatrix} Y_{11} & \cdots & Y_{1m} \\ Y_{21} & \cdots & Y_{2m} \\ \vdots & \ddots & \vdots \\ Y_{k1} & \cdots & Y_{km} \end{bmatrix} \tag{5}$$

where Y_{ij} $(i = 1, 2, \ldots, k;\ j = 1, 2, \ldots, m)$ is the new ith dimensional coordinate of the jth data point.

The covariance matrix of $\mathbf{X}$, $COV(\mathbf{X})$, is defined as

$$COV(\mathbf{X}) = \frac{1}{m-1}\mathbf{X}\mathbf{X}^{\mathbf{T}} \tag{6}$$

Then, the covariance matrix of $\mathbf{Y}$, $COV(\mathbf{Y})$, is computed as

$$COV(\mathbf{Y}) = \frac{1}{m-1}\mathbf{Y}\mathbf{Y}^{\mathbf{T}} \tag{7}$$

$$= \mathbf{V}\, COV(\mathbf{X})\, \mathbf{V}^T \quad \text{(see [17])} \tag{8}$$

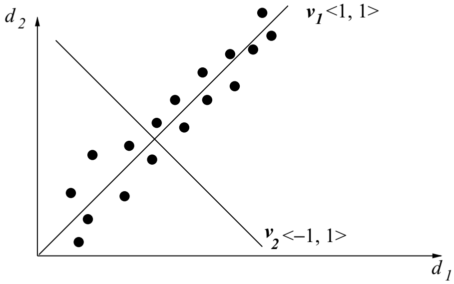


Fig. 3 An example of PCA vectors

This $COV(\mathbf{Y})$ is diagonalized when the vectors $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_k}$ in Eq. (3) are the eigenvectors of $COV(\mathbf{X})$; the variance along each vector corresponds to its eigenvalue. These eigenvectors are the PCA vectors.

As mentioned in the introduction, the last PCA vector $\mathbf{v_k}$ has the following *minimum variance* property.

Property 1 (Minimum variance) The last PCA vector $\mathbf{v_k}$ of the ranked set of PCA vectors $\mathbf{V} \equiv \langle \mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_k} \rangle$ has the minimum variance along its direction among all the vectors in a k-dimensional space.

Proof The proof of thisproperty is based on the fact that the orthonormal transformation coming from the eigenvectors minimizes the trace of the matrix $COV(\mathbf{Y})$. For details, we refer the reader to Property A2 on page 11 of Jolliffe [17]. □

Example 5 Figure 3 shows an example of the PCA vectors $\mathbf{v_1}$ (first) and $\mathbf{v_2}$ (second) found for the data points in a two-dimensional space $d_1 \times d_2$. Note that the data points have the higher variance along $\mathbf{v_1}$ (i.e., $\langle 1, 1 \rangle$) and the lower variance along $\mathbf{v_2}$ (i.e., $\langle -1, 1 \rangle$). Hence, $\mathbf{v_2}$ comes after $\mathbf{v_1}$ in the PCA result.

We call the last PCA vector the *minimum variance PCA vector* in the sequel.

3.2 Terms and concepts of PP correlations

A time series S is a finite sequence of numerical values, $a_1, a_2, \ldots, a_n$, denoted as $\langle a_1, a_2, \ldots, a_n \rangle$. We assume each a_i $(i = 1, 2, \ldots, n)$ is associate with the same time interval (e.g., day) and there is a *period* p $(p < n)$ (e.g., week) inherent in the time series. Then, S can be divided into disjoint subsequences of equal length p. That is, $S \equiv \langle S_0, S_1, \ldots, S_i, \ldots, S_{\lfloor n/p \rfloor - 1} \rangle$ where, for each $i = 0, 1, \ldots, \lfloor n/p \rfloor - 1$, $S_i \equiv \langle a_{ip+1}, \ldots, a_{ip+j}, \ldots, a_{ip+p} \rangle$. Here, each $j \in [1, \ p]$ is called an *offset* within the period p. We can map each S_i to a data point in a p-dimensional space, where each dimension corresponds to one of the p offsets. In other words, each of the p dimension variables represents each of the p offsets. We use a set of dimension variables (or a *dimensional set*), denoted as $D_p \equiv \{d_1, d_2, \ldots, d_p\}$, to represent a p-dimensional space.

Example 6 Figure 4 illustrates a time series (S) of length 9 divided into 3 subsequences (S_0, S_1, S_2) of length 3 each, given a period of 3. Each of S_0, S_1, and

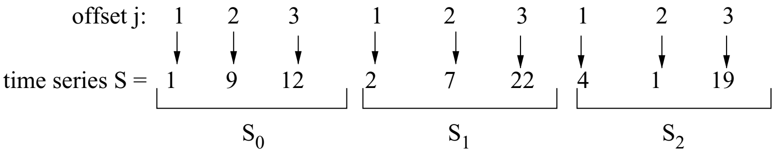


Fig. 4 A time series sequence and its subsequences

S_2 is mapped to a point in a three-dimensional space with the dimensional set $D_3 \equiv \{d_1, d_2, d_3\}$.

We are now ready to formally define PP correlations.

Definition 1 (PP correlation) Given a period p and a variance threshold V_{rmth}, a set $D_k \equiv \{d_1, d_2, \dots, d_k\}$ of offsets (or dimensions) within the period p $(k \leq p)$ is a *PP correlation* if there exists a linear constraint of the form

$$-\beta_0 + \sum_{i=1}^{k} \beta_i d_i = 0 \tag{9}$$

where $\beta_0, \beta_1, \dots, \beta_k$ are constants satisfying the following three conditions. First, $\langle \beta_1, \beta_2, \dots, \beta_k \rangle$ is a PCA vector ($\mathbf{v}$), hence

$$\sqrt{\beta_1{}^2 + \beta_2{}^2 + \cdots + \beta_k{}^2} = 1 \tag{10}$$

Second, β_0 is the mean of data points projected onto the vector $\mathbf{v}$, which is computed as

$$\beta_0 = \begin{bmatrix}\beta_1 & \beta_2 & \cdots & \beta_k\end{bmatrix} \begin{bmatrix}\bar{X}_1 \\ \bar{X}_2 \\ \vdots \\ \bar{X}_k\end{bmatrix} \tag{11}$$

Third, the variance of data points projected onto $\mathbf{v}$, $var_{\mathbf{v}}$, is lower than V_{rmth}:

$$\mathrm{var}_{\mathbf{v}} \equiv \frac{1}{(m-1)} \sum_{j=1}^{m} \left(\begin{bmatrix}\beta_1 & \beta_2 & \cdots & \beta_k\end{bmatrix} \begin{bmatrix}X_{1j} \\ X_{2j} \\ \vdots \\ X_{kj}\end{bmatrix} - \beta_0 \right)^2 < V_{\mathrm{th}} \tag{12}$$

where m is the number of data points mapped from the time series of length n within each period p i.e., $m = \lfloor n/p \rfloor$), X_{ij} $(i = 1, 2, \dots, k,\ j = 1, 2, \dots, m)$ is the ith dimensional coordinate value of the jth data point, $\bar{X}_i$ is the average value of the ith dimensional coordinate, and $var_{\mathbf{v}}$ is the variance along the vector $\mathbf{v} \equiv \langle \beta_1, \beta_2, \dots, \beta_k \rangle$.

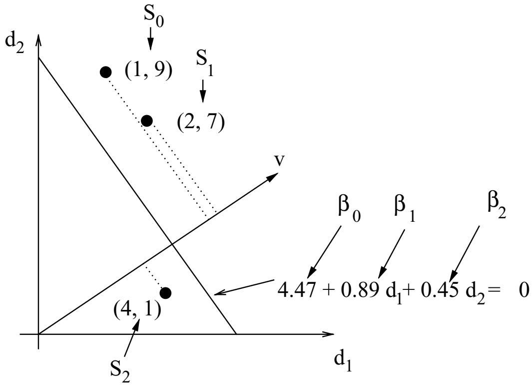


Fig. 5 A PP correlation with the two-dimensional set d_1, d_2 from the time series shown in Fig. 4

In this definition, the first condition (Eq. (10)) prevents the trivial solution $\beta_0 = \beta_1 = \cdots = \beta_k = 0$; the third condition (Eq. (12)) prevents weak correlations from being PP correlations. Geometrically, $var_{\mathbf{v}}$ is a measure of how closely Eq. (9) fits the data points. Note that, when using Eq. (9) (for predicting future values), we do not need to subtract the mean ($\bar{X}_i$) from the ith coordinate value of a data point for each dimension variable d_i ($i = 1, 2, \ldots, k$). It is compensated by subtracting the term β_0.

Example 7 Figure 5 illustrates a PP correlation with the dimensional set $D_2 \equiv \{d_1, d_2\}$, which is a subset of $D_3 \equiv \{d_1, d_2, d_3\}$ for the time series shown in Fig. 4. The three points at the coordinates $(d_1, d_2) \equiv$ (1,9), (2,7), and (4,1) are mapped from the first two offsets ($j = 1, 2$) of the subsequences S_1, S_2, and S_3 (of period 3), respectively. The linear constraint for the PP correlation is expressed as $-4.47 + 0.89d_1 + 0.45d_2 = 0$. The vector $\mathbf{v} = \langle 0.89, 0.45 \rangle$ is the one perpendicular to the line given by the linear constraint, and $var_{\mathbf{v}}$ corresponds to the variance of the three points projected to $\mathbf{v}$.

It is now easy to define minimum PP correlations as follows.

Definition 2 (Minimum PP correlation) *A PP correlation is called* minimum *if it is not the union of two or more PP correlations.*

4 Mining PP correlations using PCA

The problem addressed in this paper can be stated as follows: given a period p and a variance threshold V_{th}, find the set of all minimum PP correlations whose variances are smaller than V_{th} in all dimensional sets of size up to p (or some lower value for computational efficiency). In this section, we provide and analyze an algorithm that addresses this problem.

find_minimum(S: time series, p: period, V_{th}: variance threshold, Max_dsize: maximum size of a dimensional set)

1. From S with the period p, compute a covariance matrix $\mathbf{A}$ and a coordinate mean vector $\mathbf{D}$.
2. PP_correlation_set = { }
3. for each dimensional set size k from 1 to Max_dsize begin
4. for each dimensional set $DSet$ of size k that cannot be constructed as the union of any PP correlations found so far begin
   // See Equation 14.
5. extract from $\mathbf{A}$ the covariance matrix $\mathbf{C}_{DSet}$ for $DSet$.
6. compute PCA vectors $\mathbf{v_1}, \mathbf{v_2}, ..., \mathbf{v_k}$ for dimensional set $DSet$ using $\mathbf{C}_{DSet}$.
7. if variance along $\mathbf{v_k} < V_{th}$ then begin
8. derive from $\mathbf{D}$ and $\mathbf{v_k}$ the equation of the hyperplane, $\mathcal{E}_{DSet}$, with $\mathbf{v_k}$ as its normal vector. // See Equation 15.
9. insert $\mathcal{E}_{DSet}$ into PP_correlation_set.
10. end if
11. end for
12. end for
13. return PP_correlation_set.

Fig. 6 An algorithm for finding minimum PP correlations

4.1 Algorithm

The algorithm find_minimum, shown in Fig. 6,mines for the minimum PP correlations for a given period p. As mentioned in the introduction, we speed up the mining process by mining all the PP correlations in one pass. This is done by pre-computing the covariance matrix (see Eq. (6)) (and a "coordinate mean vector" to be described below) in a single pass and, then, extracting parts of it to find PP correlations without revisiting the time series data. In the remainder of this subsection, we first describe the algorithm—an overview and the details—and explain the correctness of the algorithm.

4.1.1 Overview

Once the pre-computations are done, the algorithm first finds all minimum PP correlations of size one. Then, it looks for minimum PP correlations with an incrementally larger dimensional set. For each dimensional set thus considered, the algorithm uses an appropriate part (to be detailed below) of the pre-computed covariance matrix and find the vector along which the data have the lowest variance. This is done using the PCA technique explained in Sect. 3.1. The key idea is to exploit the minimum variance property (Property 1) to find the PCA vector of minimum variance, $\mathbf{v_{min}}$. If the minimum variance is below the threshold V_{th}, then we have found a PP correlation, with its linear constraint being the hyperplane that uses $\mathbf{v_{min}}$ as its normal vector. The output of the algorithm is a set of linear constraints of the PP correlations. As mentioned in the introduction, such linear constraints are useful in many applications (see Examples 1, 2, and 3).

4.1.2 Details

The algorithm needs the time series data (S) to be mined, a known period (p) and also two user-provided parameters: the variance threshold (V_{th}) and the maximum PP correlation size (*Max_dsize*) considered by the algorithm. It is straightforward to the user to determine an appropriate value of Max_dsize, but it may not be so for V_{th}. In this case, the user may run a small-scale test of the algorithm with the maximum possible value of V_{th} and rank the PP correlations found from the one with the smallest variance first. This will give the user a clue to an appropriate value.

As the first step, the algorithm pre-computes two kinds of data from S and p: a covariance matrix (**A**) and a *coordinate mean vector* (**D**) of the p-dimensional data points mapped from S. **A** is computed as in Eq. (6) (with k equal to p); this results in a $p \times p$ matrix:

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1p} \\ A_{21} & A_{22} & \cdots & A_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ A_{p1} & A_{p2} & \cdots & A_{pp} \end{bmatrix} \tag{13}$$

where A_{ij} is the covariance between the ith and the jth dimensions. **D** is computed as $\langle \bar{X}_1, \bar{X}_2, \ldots, \bar{X}_p \rangle$ where $\bar{X}_i$, $i = 1, 2, \ldots, p$, is the ith coordinate mean, i.e., mean of the ith coordinate values of data points, in the p-dimensional space. These two data are used to find, for each dimensional set searched, the equation of the hyperplane with minimum variance.

Then, given the inputs V_{th} and Max_dsize as well as the pre-computed **A** and **D**, the algorithm searches through all the minimum correlated dimensional sets using a bottom-up strategy. That is, it starts from the dimensional sets of size one and increases the size by one at each iteration (Line 3). This bottom-up strategy allows for pruning the search space by not searching PP correlations that are not minimum, that is, equal to the union of some smaller PP correlations (Line 4). At each iteration, the algorithm finds the minimum PP correlation in the current dimensional set ($DSet$). For this purpose, it first extracts from the covariance matrix **A** a smaller covariance matrix ($\mathbf{C}_{DSet}$) for the dimensional set $DSet$ (Line 5). Given $DSet$ of size k, i.e., $\{d'_1, d'_2, \ldots, d'_k\} \subseteq \{d_1, d_2, \ldots, d_p\}$, $\mathbf{C}_{DSet}$ is given as

$$\mathbf{C}_{DSet} = \begin{bmatrix} A_{d'_1 d'_1} & A_{d'_1 d'_2} & \cdots & A_{d'_1 d'_k} \\ \vdots & \vdots & \ddots & \vdots \\ A_{d'_k d'_1} & A_{d'_k d'_2} & \cdots & A_{d'_k d'_k} \end{bmatrix} \tag{14}$$

where each element $A_{d'_i d'_j}$ ($i, j \in \{1, 2, \ldots, k\}$) is extracted from **A** shown in Eq. (13).

Then, given the $\mathbf{C}_{DSet}$, the algorithm finds PCA vectors $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_k}$ by computing the eigenvectors of $\mathbf{C}_{DSet}$, as mentioned in Sect. 3.1 (Line 6). Then, it takes the last PCA vector $\mathbf{v_k}$, and checks if the variance along its direction is below the threshold V_{th} to determine whether a PP correlation exists in $DSet$ (Line 7). As mentioned in Sect. 3.1, the variance along $\mathbf{v_k}$ is its eigenvalue. This is the same variance as $var_{\mathbf{v}}$ defined in Eq. (12).

If a PP correlation exists in $DSet$, then the algorithm derives the equation of the hyperplane, $\mathcal{E}_{DSet}$, representing the PP correlation. The equation involves $DSet$, i.e., $\{d'_1, d'_2, \ldots, d'_k\}$, and the minimum variance PCA vector $\mathbf{v_k}$ (denoted as $\langle \beta_1, \beta_2, \ldots, \beta_k \rangle$) as its normal vector, and can be written as follows:

$$-\sum_{i=1}^{k} \beta_i \bar{X'}_i + \sum_{i=1}^{k} \beta_i \, d'_i \;=\; 0 \tag{15}$$

where β_i is the ith component of $\mathbf{v_k}$ and $\bar{X'}_i$ is the ith coordinate mean in the k-dimensional space $d'_1 \times d'_2 \times \ldots \times d'_k$. Note that this equation is of the same form as Eq. (9). The first term is the same as Eq. (11).

4.1.3 Correctness

The following proposition states that the algorithm find_minimum is correct.

Proposition 1 *Given a dimensional set of size Max_size, the algorithm find_minimum finds all possible minimum PP correlations in the dimensional set.*

This proposition is true for the following reason. By definition, a PP correlation is said to exist in a dimensional set if the variance along the last PCA vector ($\mathbf{v_k}$) is smaller than V_{th}. Since this is the condition checked by the algorithm to find a PP correlation (in Line 7), the algorithm does find one if it exists in a given dimensional set. Moreover, the algorithm searches *every* minimum correlated dimensional set of size one to Max_size because of its bottom-up search strategy which starts with the dimensional sets of the smallest size (see Line 3). Hence, the algorithm finds all the minimum PP correlations.

Example 8 Figures 7 and 8 show examples of how PCA can be used to find PP correlations. Suppose the given period is four and $V_{\text{th}} = 1.0$. Figure 7a shows the time series S in three periods and the three four-dimensional data points (point 1, point 2, point 3) mapped from S; Fig. 7b shows the pre-computed **X**, **A**, and **D**. Figure 8a and b shows the intermediate covariance matrices and PP correlation equations computed when finding PP correlations in the dimensional sets $\{d_1, d_2\}$ and $\{d_3\}$, respectively. Specifically, Fig. 8a shows the minimum variance PCA vector $\mathbf{v_2}$ and the extracted covariance matrix $\mathbf{C}_{DSet}$ used to find the PP correlation in the dimensional set $\{d_1, d_2\}$. Figure 8b shows the same for dimensional set $\{d_3\}$.

4.2 Analysis

4.2.1 Number of passes over time series

As mentioned already, the algorithm find_minimum needs to scan the time series data only once to find all the minimum PP correlations. This is due to a combination of the following two properties. First, the algorithm computes **D** and **A** once and reuses it to find PP correlations in all candidate dimensional sets without revisiting the data. Second, the computation of **D** and **A** is done incrementally as more data points are added. The first property has been explained in the algorithm description above. The reason for the second property is as follows. It is trivial

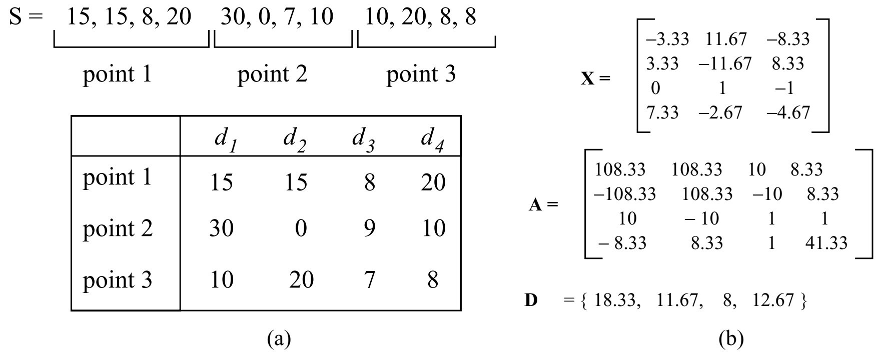


Fig. 7 An example time series S and the pre-computed **X**, **A** and, **D**

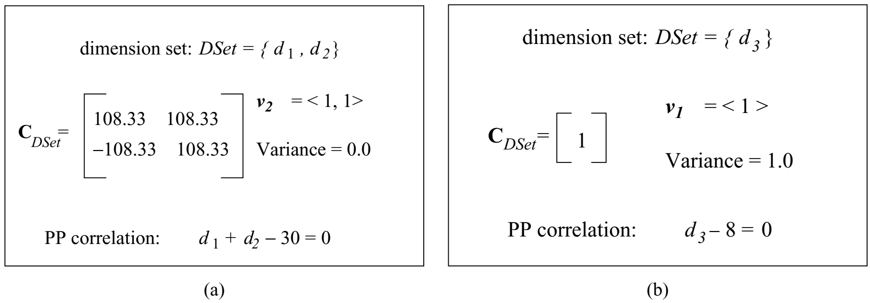


Fig. 8 Examples of applying PCA to find PP correlations in the time series S of Fig. 7

to see that **D** can be computed incrementally. For **A**, using Eq. (6), each of its elements A_{ij} is computed as

$$A_{ij} = \frac{1}{m-1} \sum_{r=1}^{m} (X_{ir} - \bar{X}_i)(X_{jr} - \bar{X}_j) \tag{16}$$

$$= \frac{1}{m-1} \left(\sum_{r=1}^{m} X_{ir} X_{jr} - \bar{X}_j \sum_{r=1}^{m} X_{ir} - \bar{X}_i \sum_{r=1}^{m} X_{jr} + \sum_{r=1}^{m} \bar{X}_i \bar{X}_j\right) \tag{17}$$

Each of the summations in the above equation can be computed incrementally. The first summation multiplies two dimensions always at the same rth data point and, therefore, the data point can be discarded once used to update the summation incrementally. The second summation can be computed incrementally since the average of the data values in the jth dimension ($\bar{X}_j$) can be computed incrementally and the sum of the ith components of all the data points can also be computed incrementally. Following this argument, the third summation can also be computed incrementally. Finally, the fourth summation can be computed incrementally since it equals m multiplied by the two averages $\bar{X}_i$ and $\bar{X}_j$ (which can both be computed incrementally).

4.2.2 Memory required

The amount of memory required to find the PP correlations is the sum of the amounts of memory needed to store the **A** matrix (whose size is proportional to p^2), the coordinate mean vector **D** (whose size is proportional to p) and the PP_correlation_set (resulting from the algorithm find_minimum). Thus, the amount of memory required(MR) can be given as

$$MR(p) \; = \; p^2 + 2p + CM \tag{18}$$

where CM denotes the amount of memory for storing the PP_correlation_set. Note that MR is shown as a function of p to reflect the p^2 term (for storing **A**) and the $2p$ term (one p for storing **D** and another p for storing the sum of each dimension needed during the computation of **A**).

4.2.3 Run time complexity

The time complexities of pre-computing a covariance matrix **A** and a coordinate mean vector **D** (Line 1) are as follows. Computing the covariance matrix **A** involves the computation shown in Eq. (6); its complexity is $O(mp^2)$ where p is the period and m is the number of data points. Computing **D** involves only averaging the values of coordinates in each dimension; therefore, its complexity is $O(pm)$. Thus, the total pre-computation complexity is $O(mp^2)$.

For the rest of the algorithm, the worst-case run time will be exponential to $min(p, Max\_dsize)$ since we may need to consider all subsets of up to size p or Max_dsize, whichever is smaller. Note this complexity is independent of m (the number of data points). The average time complexity in practice, however, is not as bad, as we will show in our experimental results.

5 Evaluations

We have evaluated the efficiency and efficacy of our PCA-based technique through three experiments. The first experiment regards the effects of the following parameters on the execution time: the number of data points (m) (i.e., the number of periods considered), the number of dimensions (i.e., the period p), and the maximum size of a dimensional set (Max_dsize). Naturally, we use synthetic time series for this experiment. The second experiment regards comparing the number of PP correlations found between mining minimum PP correlations and mining all PP correlations. We use both synthetic and real time series for this experiment. The third experiment regards the actual minimum PP correlations found in real time series. In this section, we first describe the experimental setup and then present the experimental results.

5.1 Experimental setup

Variants of the PCA-based technique: To demonstrate different aspects of our technique, we have generated three variants of the algorithm *find_minimum*. The first one is *find_minimum_multipass*, which is the same as find_minimum (see Fig. 6) with the exception that, in Line 5 of the algorithm, the one-pass covariance computation is disabled so that each covariance matrix $\mathbf{C}_{DSet}$ is computed directly from the time series data instead of being extracted from **A**. The second one is *find_all*, which finds all PP correlations (i.e., including non-minimum PP correlations) using the same algorithm as find_minimum except that, in Line 4 of the algorithm, all the dimensional sets of size $dsize$ are searched. The third one is *find_all_multipass*, which is a combination of the first two variants.

Synthetic data sets: We have built a synthetic time series generator which works as follows. It needs two input parameters: the period (p) and the number of data points (m). It first generates $2p - 2$ potentially overlapping dimensional sets, consisting of two-dimensional sets of size one, two-dimensional sets of size two, and so on, up to two-dimensional sets of size $p - 1$. Then, it generates a time series running for m periods while introducing one PP correlation using each of the $2p - 2$ dimensional sets generated. The domain of the value of a time series element is a real number in the range of 0 to 100. To add some variance to the data, we add to each element value a random number with the uniform distribution between 0 and 1.0% of the value.

Real data sets: We have used two different real data sets. The first one is a time series of hourly residential power consumption over a period of 1 year. It is the same time series as the one used in [8]. The second one is a time series of daily (weekdays only) closing price of IBM stock from January 1, 1980 to October 8, 1992 [14].

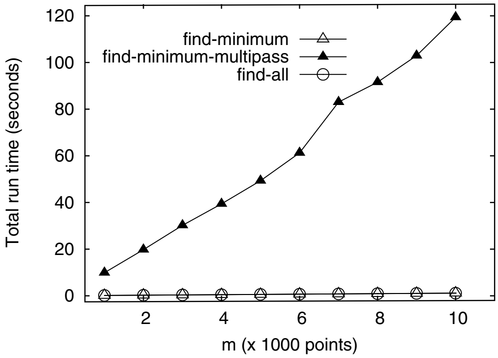
Computing platform: All the experiments were performed on a laptop computer with a single 1.8 GHz Intel Centrino CPU, 1 GB RAM, and 60 GB hard disk.

5.2 Experimental results

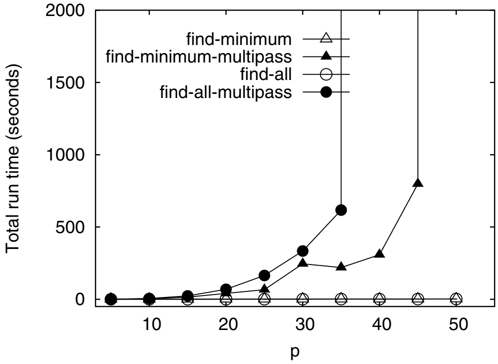
5.2.1 Experiment 1: execution time for varying parameter values

Figure 9 shows the execution time of *find_minimum*, *fund_minimum_multipass*, *find_all*, and *find_all_multipass* measured while varying the value of each of the three parameters (m, p, Max_dsize) with the other two set to the medians of their respective ranges.

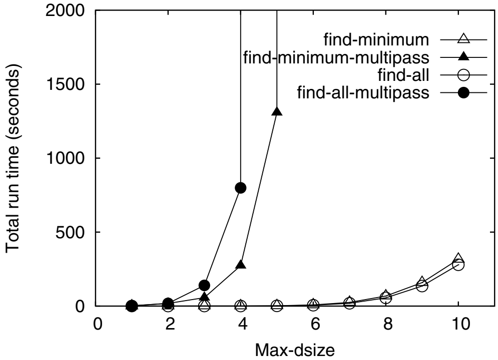
Figure 9a shows the execution times of the algorithms for a varying number of data points. (The curve for find_all_multipass does not show in the figure because the execution time exceeds the maximum value that can be shown in the figure (2000 s) for even the smallest number of data points used in the figure (1000 points).) The figure shows that one-pass algorithms (i.e., find_minimum, find_all) scale much better with the number of data points than the multipass counterparts (i.e., find_minimum_multipass, find_all_multipass). This is as expected, because repeated passes through the time series data become increasingly more costly due to an increase in the PCA computation cost per scan as the number of data points (i.e., time series periods scanned) increases. Since find_minimum and find_all take only one pass through the time series data, their relative performances are largely unaffected by the varying number of data points.



(a)



(b)



(c)

Fig. 9 Execution time for varying values of the parameters ($V_{\text{th}} = 1$). **a** Varying number of data points (m) ($p = 24$, $Max\_dsize = 3$). The curve of find_all_multipass is out of bound. **b** Varying period (p) ($m = 5000$, $Max\_dsize = 3$). **c** Varying maximum dimension-set size (Max_dsize) ($m = 5000$, $p = 24$)

Figure 9b shows the execution times of the four algorithms for a varying period. (The curves for find_all_multipass and find_minimium_multipass do not show the points at the periods 35 and 45 respectively, or larger because the execution times at those periods exceed the maximum value that can be shown in the figure (2000 s).) The figure shows that one-pass algorithms are *increasingly* more efficient than multipass algorithms as the period increases. The reason is that the number of dimensional sets searched increases with the increasing period and, therefore, so does the overhead of scanning the time series for all the dimensional sets.

Figure 9(c) shows the execution times of the four algorithms for a varying maximum dimension-set size (Max_dsize). The curves show the same trend as those in Fig. 9(a) and (c). Interestingly, find_all is slightly faster than find_minimum when Max_dsize is large. This happens when find_minimum's overhead of checking if a dimensional set equals the union of any PP correlations already found is higher than find_all's overhead of looking for all PP correlations (in one scan).

5.2.2 Experiment 2: the number of PP correlations found

In this experiment, we use only the one-pass algorithms since the number of PP correlations found is independent of the number of passes. We use both synthetic and real data sets for this experiment. Figure 10 shows the results for varying the variance threshold V_{th} while setting Max_dsize to 5 for all the data sets and setting the period to 24 for the synthetic or power consumption data set and to 5 for the IBM stock price data set. The figure shows that the number of PP correlations found using find_minimum are orders of magnitude smaller than that found using find_all. This demonstrates how effective the pruning of find_minimum is.

The result of find_all for the power consumption data set (Fig. 10b) shows a different trend from those for the synthetic (Fig. 10a) and IBM stock price (Fig. 10c) data sets. The difference is that the number of PP correlations found decreases dramatically as V_{th} decreases for the power consumption data set whereas it stays relatively constant for the other data sets. The reason for this is that the power consumption data have higher variance than data in the other data sets and, therefore, far fewer PP correlations can be found at a lower variance threshold(V_{th}).

In the find_minimum result, sometimes the number of PP correlations found decreases with an increasing threshold value. This happens if, with a higher threshold value, more PP correlations are found in small dimensional sets but they are used to prune out even more PP correlations from larger dimensional sets. This phenomenon is more likely to happen with the IBM stock price data set because the number of possible PP correlations is far smaller (i.e., 26 (= $\sum_{i=2}^{5} {}_5C_i$) instead of 16,777,191 (= $\sum_{i=2}^{24} {}_{24}C_i$)) given the shorter period (i.e., 5 instead of 24).

5.2.3 Experiment 3: PP correlations found in real data sets

Table 1 shows, given different values of the variance threshold (V_{th}), the total number of PP correlations found in real data sets and the example minimum PP correlations.

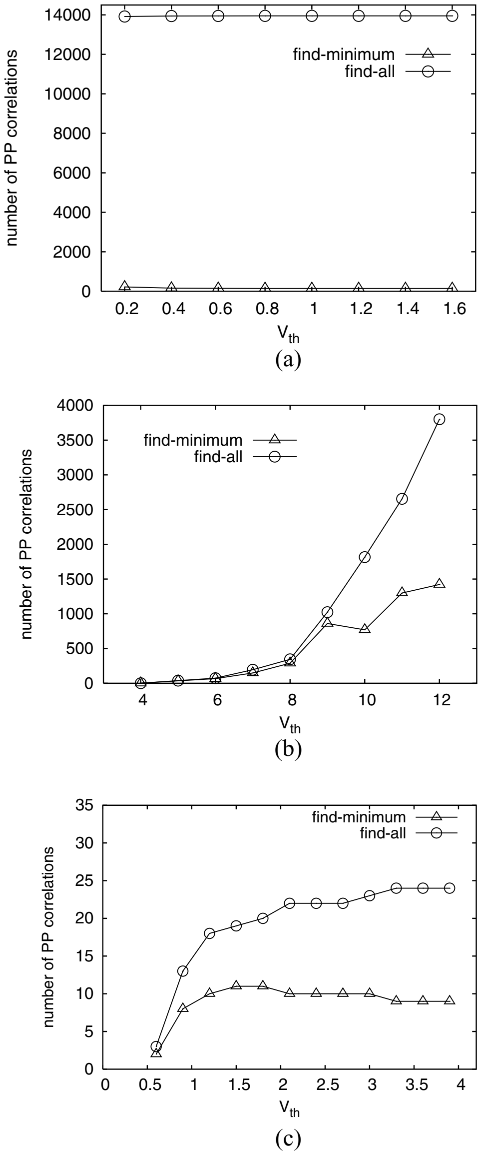


Fig. 10 The number of PP correlations found for varying V_{th}. **a** Synthetic data set (period=24, $Max\_dsize = 5$, $m = 5000$). **b** Power consumption data set (period=24, $Max\_dsize = 5$). **c** IBM stock price data set (period=5, $Max\_dsize = 5$)

Table 1a shows the results for the power consumption data set. The period used is 24 (hours), and the dimensions $d_1, d_2, \ldots, d_{24}$ correspond to the 24 h of the day, i.e., 12 a.m., 1 a.m., …, 11 p.m. Interestingly, we have observed that PP correlations with low variances (with variances below 6) are found among the dimensions d_1 through d_8. Since these dimensions correspond to early morning hours (12 a.m.–7 a.m.), we infer that the correlation exists because there is not

Table 1 Minimum PP correlations found in real time series data sets

V_{th}	Number of minimum PP correlations	Example minimum PP correlation
5.0	35	$-1.671 + 0.426d_2 - 0.815d_3 + 0.394d_4 = 0$
6.0	66	$+2.246 - 0.332d_2 - 0.717d_3 - 0.569d_4 - 0.21d_5 - 0.022d_8 = 0$
7.0	148	$3.115 - 0.567d_2 - 0.796d_3 - 0.213\, d_5 - 0.011\, d_{14} + 0.01\, d_{21} = 0$
8.0	289	$-1.708 - 0.128d_{15} + 0.559d_{16} - 0.735d_{17} + 0.359d_{18} - 0.05d_{20} = 0$
	(a) Power consumption data set. ($p = 24$, $Max\_dsize = 5$, $\{d_1, d_2, \ldots, d_{24}\} = \{$12 a.m., 1 a.m., ..., 11 p.m.$\}$)	
V_{th}	Number of minimum PP correlations	Example minimum PP correlation
0.6	2	$0.1558 - 0.239d_2 + 0.704d_3 - 0.645d_4 + 0.178d_5 = 0$
1.0	9	$0.081 + 0.387d_1 - 0.816d_2 + 0.439d_3 = 0$
1.4	10	$-0.074 + 0.707d_1 - 0.707d_2 = 0$
1.8	11	$0.115 + 0.516d_1 - 0.806d_3 + 0.29d_5 = 0$

(b) IBM stock price data set.
($p = 5$, $Max\_dsize = 5$, $\{d_1, d_2, \ldots, d_5\} = \{$Monday, Tuesday, ..., , Friday$\}$)

much human activity during these early, sleeping hours and, consequently, the power consumption is stable, limited to refrigerators, hot water furnaces, etc. that consume power with little fluctuation.

Table 1b shows the results for the IBM stock price data set. The period used is 5 (days), and the dimensions $d_1, d_2, \ldots, d_5$ correspond to the weekdays, i.e., Monday, Tuesday, ..., Friday. Although we know that the data set contains cyclic correlations conforming to the definition of PP correlations, it is hard for us to identify the *cause* of the PP correlations found, due to our insufficient domain knowledge in IBM stock price data. Such a cause analysis is beyond the scope of this paper anyway.

6 Extension to multiple periods

In this section, we discuss a simple extension of our technique, by which multiple periods in a given range are considered in one pass. Consider a range of periods $p_l, p_{l+1}, \ldots, p_h$, and denote the covariance matrix and the coordinate mean vector computed for a given p_k $(l \leq k \leq h)$ as $\mathbf{A}(p_k)$ and $\mathbf{D}(p_k)$, respectively. Then, since each $\mathbf{A}(p_k)$ and $\mathbf{D}(p_k)$ can be computed incrementally, we can compute all the covariance matrices $\mathbf{A}(p_l), \mathbf{A}(p_{l+1}), \ldots, \mathbf{A}(p_h)$ and all the coordinate mean vectors $\mathbf{D}(p_l), \mathbf{D}(p_{l+1}), \ldots, \mathbf{D}(p_h)$ in one pass. Then, for each p_k $(l \leq k \leq h)$, the pair of $\mathbf{A}(p_k)$ and $\mathbf{D}(p_k)$ can be passed to the algorithm find_minimum to find the minimum PP correlations.

Since we pre-compute all the $\mathbf{A}(p_k)$ and $\mathbf{D}(p_k)$ for $i \leq k \leq j$, the number of passes required is still only one. Given the range of periods p_l–p_h, the amount of memory required, $MRR(p_l, p_h)$, is given by the following equation.

$$MRR(p_l, p_h) = \sum_{k=l}^{h}(p_k^2 + 2p_k) + CMR \tag{19}$$

where *CMR* denotes the amount of memory for storing the PP correlations found for all periods in the given range. Similarly to Eq. (18), the p_k^2 term is for storing the pre-computed $\mathbf{A}(p_k)$ and the $2p_k$ term is for storing the pre-computed $\mathbf{D}(p_k)$ and the sum of each dimension needed during the computation of $\mathbf{A}(p_k)$.

7 Conclusions

In this paper, we introduced the notion of partial periodic (PP) correlations in time series, which are shown in our experiments to appear frequently in real data. For mining the PP correlations, we developed a PCA-based technique. In order to avoid returning too many PP correlations, we focused on finding the *minimum* PP correlations. In addition, we developed our technique in a way that it only needs to have *one pass* through the time series data. Finally, we used experiments involving both real and synthetic data sets to demonstrate that our approach is computationally efficient and do find PP correlations existing in real time series.

Two research subjects may be of interest for some further study. The first area is to use our PCA-based technique to mine PP correlations in *data streams*. This is possible since the covariance matrix can be computed incrementally (as explained in Sect. 4.2). Our technique can then use the matrix to mine PP correlations on demand without revisiting the data. The second area is to further develop the approach described in Sect. 6 to create a more efficient way of mining PP correlations for a range of periods. One approach may be to reuse some information (e.g., co-variance matrix) of one period to avoid or accelerate the computation of PP correlations for a different period.

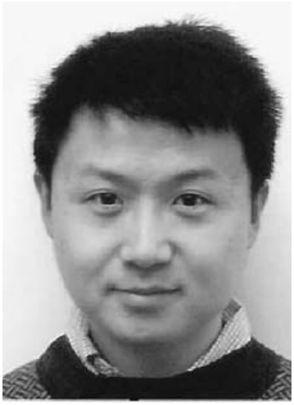
Acknowledgements We thank Mohamed Elfeky at Purdue University for providing us with the power consumption time series data set. This research has been supported by the National Science Foundation through Grant No. IIS-0415023 and the US Department of Energy through Grant No. DE-FG02-ER45962.

References

1. Aref W, Elfeky M, Elmagarmid A (2004) Incremental, online, and merge mining of partial periodic patterns in time-series databases. IEEE Trans Knowledge Data Eng
2. Berberidis C, Aref W, Atallah M, Vlahavas I, Elmagarmid A (2002) Multiple and partial periodicity mining in time series databases. In: Proceedings of the European conference on artificial intellegence, pp 370–374
3. Box P, Jenkins M, Reinsel C (1994) Time series analysis, forecasting and control. Prentice-Hall
4. Brown GR (1963) Smoothing, forecasting and prediction. Prentice-Hall
5. Chakrabarti K, Mehrotra S (2000) Local dimensionality reduction: a new approach to indexing high dimensional spaces. In: Proceedings of VLDB, pp 89–100
6. Chatfield C, Yar M (1988) Holt–Winters forecasting: some practical issues. The Statistician, pp 129–140

7. Cui B, Ooi BC, Su J, Tan K-L (2003) Contorting high dimensional data for efficient main memory KNN processing. In: Proceedings of ACM-SIGMOD
8. Elfeky M, Aref W, Elmagarmid A (2004) Using convolution to mine obscure periodic patterns in one pass. In: Proceedings of EDBT, pp 605–620
9. Farnum RN, Stanton WL (1989) Quantitative forecasting methods. PWS-Kent
10. Fox J (1997) Applied regression analysis, linear models, and related methods. Sage Publications, Thousand Oaks, CA
11. Gardner SE, McKenzie E (1985) Forecasting trends in time series. Manage Sci, pp 1237–1246
12. Han J, Dong G, Yin Y (1999) Efficient mining of partial periodic patterns in time series database. In: Proceedings of ICDE, pp 106–115
13. Han J, Gong W, Yin Y (1998) Mining segment-wise periodic patterns in time-related databases. In: Proceedings of international conference in knowledge discovery and data mining, pp 214–218
14. Hipel K, McLeo AI (1994) Time series modelling of water resources and environmental systems. Elsevier
15. Hui J, Ooi BC, Shen H, Yu C (2003) An adaptive and efficient dimensionality reduction algorithm for high-dimensional indexing. In: Proceedings of ICDE, pp 87–99
16. Indyk P, Koudas N, Muthukrishnan S (2000) Identifying representative trends in massive time series data sets. In: Proceedings of VLDB, pp 363–372
17. Jolliffe I (1986) Principal component analysis. Springer-Verlag
18. Ma S, Hellerstein J (2001) Mining partially periodic event patterns with unknown periods. In: Proceedings of ICDE, pp 205–214
19. Ord KJ, Koehler BA, Snyder DR (1997) Estimation and prediction for a class of dynamic nonlinear statistical models. J Am Stat Assoc 92:1621–1629
20. Shlens J (in press) A tutorial on principal component analysis. http://www.snl.salk.edu/ shlens/pub/notes/pca.pdf
21. Wang W, Yang J, Yu P (2001) Meta-patterns: revealing hidden periodic patterns. In: Proceedings of ICDM, pp 550–557
22. Yang J, Wang W, Yu P (2001) Infominer: mining surprising periodic patterns. In: Proceedings of international conference in knowledge discovery and data mining, pp 395–400
23. Yang J, Wang W, Yu P (2004) Discovering high order periodic patterns. Knowledge Inf Syst J (KAIS) 6(3):243–268
24. Yang J, Wang W, Yu PS (2000) Mining asynchronous periodic patterns in time series data. In: Proceedings of international conference in knowledge discovery and data mining, pp 275–279

Author Biographies



Zhen He is a lecturer in the Department of Computer Science at La Trobe University. His main research areas are database systems optimization, time series mining, wireless sensor networks, and XML information retrieval. Prior to joining La Trobe University, he worked as a postdoctoral research associate in the University of Vermont. He holds Bachelors, Honors and Ph.D degrees in Computer Science from the Australian National University.

X. Sean Wang received his Ph.D degree in Computer Science from the University of Southern California in 1992. He is currently the Dorothean Chair Professor in Computer Science at the University of Vermont. He has published widely in the general area of databases and information security, and was a recipient of the US National Science Foundation Research Initiation and CAREER awards. His research interests include database systems, information security, data mining, and sensor data processing.

Byung Suk Lee is associate professor of Computer Science at the University of Vermont. His main research areas are database systems, data modeling, and information retrieval. He held positions in industry and academia: Gold Star Electric, Bell Communications Research, Datacom Global Communications, University of St. Thomas, and currently University of Vermont. He was also a visiting professor at Dartmouth College and a participating guest at Lawrence Livermore National Laboratory. He served on international conferences as a program committee member, a publicity chair, and a special session organizer, and also on US federal funding proposal review panel. He holds a BS degree from Seoul National University, MS from Korea Advanced Institute of Science and Technology, and Ph.D from Stanford University.

Alan C. H. Ling is an assistant professor at Department of Computer Science in University of Vermont. His research interests include combinatorial design theory, coding theory, sequence designs, and applications of design theory.