Multivariate Time Series Representation and Similarity Search Using PCA

Aminata Kane and Nematollaah Shiri

Computer Science and Software Engineering
Concordia University, Montreal, Canada
{am_kane,shiri}@cse.concordia.ca

Abstract. Multivariate time series (MTS) data mining has attracted much interest in recent years due to the increasing number of fields requiring the capability to manage and process large collections of MTS. In those frameworks, carrying out pattern recognition tasks such as similarity search, clustering or classification can be challenging due to the high dimensionality, noise, redundancy and feature correlated characteristics of the data. Dimensionality reduction is consequently often used as a preprocessing step to render the data more manageable. We propose in this paper a novel MTS similarity search approach that addresses these problems through dimensionality reduction and correlation analysis. An important contribution of the proposed technique is a representation allowing to transform the MTS with large number of variables to a univariate signal prior to seeking correlations within the set. The technique relies on unsupervised learning through Principal Component Analysis (PCA) to uncover and use, weights associated with the original input variables, in the univariate derivation. We conduct numerous experiments using various benchmark datasets to study the performance of the proposed technique. Compared to major existing techniques, our results indicate increased accuracy and efficiency. We also show that our technique yields improved similarity search accuracy.

Keywords: Multivariate time series · Dimensionality reduction · Similarity search

1 Introduction

Innovation and advances in technology have led to the growth of data at a phenomenal rate. Paradoxically, the existing MTS data reduction, analysis and mining techniques do not scale well to its current challenges. Among those challenges, the high dimensionality of the data both in terms of the number of variables and the length of the time series, but also the presence of noise and redundancies makes it difficult to uncover important patterns for many practical applications.

Hence, most pattern recognition tasks rely on dimensionality reduction as a crucial preprocessing step, for reasons of efficiency and interpretability, for a

P. Perner (Ed.): ICDM 2017, LNAI 10357, pp. 122–136, 2017.
DOI: 10.1007/978-3-319-62701-4_10

better understanding of the underlying processes that generated the data, but also to afford a framework that allows downstream pattern recognition tasks to perform more efficiently.

The adequacy of the chosen reduction technique is very important as it will greatly affect the overall quality of search. In similarity search for instance, MTS reduction techniques often follow one of three approaches. In the first approach, each variable is considered independently as a time series [6]. While being easier to process, this approach often requires much more computation time. The second approach consists of concatenating all data contained within all variables as a long univariate time series (UTS) [13]. Like the first approach, this often overlooks the relationships that exist among the variables and cannot efficiently process a relatively large number of variables. The third approach transforms the MTS into a lower dimensional representation that still captures its main characteristics while rendering the data more manageable. Although this approach presents more complexity, it provides more accurate results for the similarity search.

In this paper we propose a similarity search technique based on dimensionality reduction and time series correlations analysis. An important aspect for this technique is the proposed representation based on PCA that allows to transform the MTS with large number of variables to a UTS prior to seeking correlations. This is particularly important because, on one hand, the representation takes into account the correlation between variables within each multivariate dataset, in addition to decreasing redundancy and noise and, reducing the intrinsic high dimensionality. Other proposed univariate representations are often not able to retain the correlation between variables within each multivariate dataset [6]. On the other hand substantial research and progress in making UTS pattern recognition tasks in general, and similarity search in particular, very efficient on large datasets has occurred in recent years [23, 26, 4, 20]. The proposed representation will allow efficient UTS techniques to be easily extended to MTS.

In what follows, we formulate the problem in Section 2, review the related work in Section 3, and provide preliminaries in Section 4. Our proposed technique is introduced in Section 5. Section 6 presents our experimental results. Concluding remarks and future directions are presented in Section 7.

2 Problem Formulation

A UTS X $= < x_1, x_2, ..., x_n >$, of dimension n is a sequence of real values for a variable measured at n different timestamps. A MTS $A_{n,m}$ of n instances for m variables can be represented as a $n \times m$ matrix A (shown below) in which $a_{i,j}$ is the value of variable $X_{*,j}$ measured at time-stamp i, for $1 \leq i \leq n$, $1 \leq j \leq m$.

$$A_{n,m} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,m} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,m} \end{bmatrix}$$

We are interested in the problem of similarity search in MTS defined as follows:

Definition 1. *(MTS similarity search)*
Let $D = \{A^1_{n,m}, A^2_{n,m}..., A^q_{n,m}\}$ *be a set of MTS, each of which containing* n *instances and* m *variables; and* ϵ *be a user specified threshold value. A MTS similarity search retrieves all pairs of times series* A^i *and* A^j *in* D *such that their correlation distance does not exceed* ϵ*, for* $1 \leq i, j \leq q$*.*

Similarity search techniques in time series can be classified in two categories: subsequence search and whole sequence search. In this paper, our focus is on whole sequence search and we use Pearsons product-moment coefficient [21] as the measure to assess similarity between two time series.

3 Related Work

Transforming MTS into lower-dimensional time-series have had some interest and many dimensionality reduction methods have been proposed. Those broadly adopted in the literature include Independent Component Analysis (ICA) [5, 8], Random Projection (RP) [3, 7], and Principal Component Analysis(PCA) [27, 3, 7].

The Independent Component Analysis technique allows to find a new basis in which to represent the multivariate data. It can be considered a generalization of the PCA technique since the latter can be used as a preprocessing step in some ICA algorithms. However, while the goal in PCA in is to capture the maximum variance of data or minimize reconstruction error, the goal of ICA is to minimize the statistical dependence between the basis vectors. ICA however presents limitations that include the inability to determine the order of the independent components and the need for the input time-series data to have non-Gaussian distribution.

The Random Projection technique relies on projecting and embedding the multivariate data onto a lower dimensional subspace, randomly. It is based on the Johnson - Lindenstrauss lemma proposed in 1984 [10]. The lemma states that, given a set of points in a high-dimensional space, they can be projected and embedded into a lower dimensional subspace, such that, the distances between the points are nearly preserved. For random projections, the lower dimensional subspace is randomly chosen based on some distribution and, we can seek to have a probabilistic guaranty that the distance between two time series in the higher dimensional space will have some sort of correspondence with the distance between the same two time series in the lower dimensional space. This data reduction technique is efficient for frameworks with a relatively small number of very long time series due to the fact that, the data size k resulting from the reduction does not depend on the length of the time series but rather the number of time series [31]. It is however known to be less effective than PCA for severe dimensionality reduction [7].

The PCA technique is an orthogonal linear transformations in which one assumes all basis vectors to form an orthonormal matrix. It projects the original dataset in a new coordinate system where the directions are pairwise orthonormal. A main advantage of PCA in our work is that it guaranties the uncovering of an optimal new embedding with minimal approximation error, and hence retains the crucial underlying structure of the original data. In addition to reducing dimensionality, the transformation decreases redundancy and noise, highlights relationships between the variables and reveals patterns by compressing the data while expressing it in such a way that highlights their similarity and dissimilarity. In addition, if two MTS are similar, their PCA representations will also be similar [15]. Many similarity search techniques [28, 27, 2, 15] have relied on PCA for MTS processing as it is known to be one of the most efficiently computable techniques and a powerful tool of choice in high dimensional data environments for linear dimensionality reduction. PCA is however also limited by the fact that, as a new set of features is generated, the reduced form of the data is still a matrix. Retaining the first principal component in order to vectorize the data has been explored with some level of success in the literature [27]. However, since principal components carry in decreasing order portions of the explained variance from the data, in order to retain enough information in the new representation, one would need to retain at least a few principal components in most cases. Hence the reduced form of the data would remain in a matrix form.

4 Preliminaries

In this section we review some background, definitions and notions needed in later sections.

4.1 Notation

This section provides the notation used in this paper, if not specified otherwise.

- D denotes the set of multivariate time series(or, D' if normalized).
- D^U denotes the set of UTS resulting from the STEP1 reduction.
- $A_{n,m}$ is the multivariate time series with n instances and m variables.
- A is such that A = $[a_{ij}]$ is a matrix representing the multivariate time series.
- A^T is the matrix transpose of A.
- V is the right eigenvector matrix of size m×m, V_k is the right eigenvector matrix of size m×k.
- S denotes the diagonal matrix of the singular values of A, S_k is the diagonal matrix of the k largest singular values of A.
- $[a_i]$ is the column vector of the matrix.
- $[a]_{i,*}$ denotes the i-th row of the matrix.
- $[a]_{*,j}$ denotes the j-th column of the matrix.
- $[a]_{i,j}$ denotes the element entry at the i-th row and j-th column of the matrix.
- θ is the cumulative explained variance in the data that are represented within k retained principal components.
- ρ is the Pearson correlation coefficient.
- ϵ is the user specified correlation threshold.

4.2 Principal Component Analysis

Carrying out PCA on raw MTS data often requires some preprocessing such as mean-centering and scaling to adjust values measured on different scales to a relatively common scale, since PCA is a variance maximizing exercise. The technique is often explained through the original data matrix's covariance matrix (A^TA) eigen-decomposition. It however can also be performed through the Singular Value Decomposition(SVD) of the data matrix. In this paper, we consider the latter, framed as per below.
Let $A_{n,m}$ be a matrix with n instances and m variables, and k be the dimension of the space in which we wish to embed the data. Using a Singular Value Decomposition of the matrix, PCA returns the top k left and top k right singular vectors of A. It subsequently projects the original data on the k-dimensional subspace spanned by the chosen column singular vectors.

Definition 2. *(Singular Value Decomposition) Let A be a $n \times m$ data matrix with r as its rank. The singular value decomposition (SVD) of A is the factorization $A = USV^T$, where:*

- *U is a column-orthonormal $n \times r$ matrix whose columns are the eigenvectors of AA^T,*
- *S is a diagonal $r \times r$ matrix of the singular values s_i for A, otherwise related to the eigenvalues λ_i of the covariance matrix A^TA by $\lambda_i = s_i^2/(n-1)$, where $\lambda_1 \geq \ldots \geq \lambda_r \geq 0$, and,*
- *V is a column-orthonormal $r \times m$ matrix whose columns are the eigenvectors of A^TA.*

Often, a rank-k approximation of the dataset works well because many datasets occurring in practice present a structure that leads to only the first few principal components being non-negligible.

To identify the number of principal components to retain from each MTS, we use the relative percentage variance criterion [11] to translate the amount of variance we wish to retain in the data to the number of principal components. The number k of relevant principal components may vary for different MTS, consequently k_{max}, representing the largest of all identified ks, are to be retained. Algorithm 1 summarizes the steps in uncovering k_{max}.

5 The Proposed Technique

Given a set of MTS D= $\{A^1_{n,m}, A^2_{n,m} \cdots, A^q_{n,m}\}$ and a user specified correlation threshold ϵ, our goal is to identify all pairs of time series whose Pearson correlation value is no less than ϵ. The proposed technique follows a two-steps resolution process. It first uses a novel transformation technique (M2U) to transform the MTS to a UTS, then seeks pairwise correlations within the set of newly generate univariate series, using the Pearson product moment correlation. An important aspect about the proposed representation resulting from the M2U transformation is that, it allows efficient UTS pattern recognition techniques to be easily extended to MTS.

Algorithm 1 - Find k_{max} the number of PCs to retain

Input: $D' = \{A^1_{n,m}, A^2_{n,m} ..., A^q_{n,m}\}$ a set of normalized MTS, θ cumulative variance to retain from each MTS.
Output: The number k_{max} of principal components to retain from each MTS s.t. $k_{max} = max(k_1, k_2, ..., k_q)$

```
begin
  $k_{max} \leftarrow 0$
  for $i \leftarrow 1$ to $q$ do
      Uncover fraction of total explained variance
        f(k) $\leftarrow \Sigma^k_{z=1}\lambda_z / \Sigma^r_{z=1}\lambda_z$ for all z = {1, . . . , r}
      Choose the smallest k so that f(k) $\geqslant \theta$ and retain that
        number of k eigenvectors to keep explained variance $\theta$
        in the new embedding.
      if $k > k_{max}$ then $k_{max} \leftarrow k$
  end for
  return $k_{max}$
end
```

5.1 M2U : Multivariate Time Series to a Univariate Time Series Transformation

In this section, we formally define the transformation process then describe its underlying intuition. Algorithm 2 provides the transformation steps from line 2 to 13.

Definition 3. *(M2U (Multivariate to Univariate Transformation)). Given the matrix $A \in R^{n \times m}$ with rank $r = rank(A)$ s.t. $r \leqslant min\{n, m\}$ and $k \leqslant r$. Let V_k be the matrix containing the top k right singular vectors of A, and S_k be the matrix containing the top k singular values of A. Then, the (rank-k) univariate representation of A is defined as*
$[U_{n,1}]^k_i = \Sigma^m_{v=1} a_{i,v}\hat{w_v}$ *, for $i = 1, 2, ..., m$ where:*

- *$a_{i,v}$ is the element of matrix A at row i, and column v.*
- *$\hat{w_j} = \Sigma^k_{z=1} w_z e_{j,z}$, for $j = \{1, 2, ..., m\}$ is the weight of the column variable j within the given multivariate dataset, called weighted score and below defined.*
- *$[U_{n,1}]^k_i = \Sigma^m_{v=1} a_{i,v}\hat{w_v}$ is the i-th entry of the newly generated UTS $U_{n,1}$.*

We assume that each MTS $A^i_{n,m}$ of n instances for m variables within D can be represented as a $n \times m$ normalized matrix A (shown below).

$$\begin{array}{cc} & \begin{array}{cccc} X_{*,1} & X_{*,2} & \cdots & X_{*,m} \end{array} \\ A_{n,m} = & \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,m} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,m} \end{bmatrix} \end{array}$$

Each column variable $X_{*,j}$ holds a particular weight or importance $\hat{w}_j$ with respect to the whole data matrix $A_{n\times m}$ [14]. Let us consider $\hat{w}$ the weight vector containing all variable weights. Intuitively, if we seek to transform the MTS to a UTS in a new framework, there will be a need to uncover and take into account the variable's importance or weight in the reconstruction process.

Finding the Weighted Scores(Variable Weights) We rely on unsupervised learning through a principal component analysis of the input data to uncover the variable weights (weighted scores) within $\hat{w}$. We use information drawn from the diagonal of the matrix S and the rows of matrix V (from the factorization $A = USV^T$) to computed statistics that reveal influence on the columns of the original matrix A.

Let us first note that the entries in each column of $V = A^T US^\dagger$ (where $S^\dagger$ denotes the Moore pseudo-inverse of S) provide the regression coefficients of a corresponding principal component, which in turn is expressed as a linear combination of all variables from the original matrix. More precisely, the coefficient of the i^{th} new feature component uncovered through PCA is expected to be the i^{th} entry of the eigenvector. The first k principal components can be expressed as below illustrated if we consider $X_1, ..., X_m$ to be the original variables within the data matrix A.

$$\begin{aligned}
&e_{1,1}X_1 + e_{1,2}X_2 + e_{1,m}X_m = PC_1\\
&e_{2,1}X_1 + e_{2,2}X_2 + e_{2,m}X_m = PC_2\\
&\dots\\
&e_{k,1}X_1 + e_{k,2}X_2 + e_{k,m}X_m = PC_k
\end{aligned}$$

Just as the principal components can be expressed as a linear combination of all variables from the original matrix, the original variables can also be defined as linear combinations of the principal components. The rows of V hence each concern a specific variable and are considered rescaled data projected onto the principal components; the data is indeed rescaled according to the singular values to ensure that the covariance is identity.

In the multivariate to univariate transformation process, we wish to uncover the influence of the original variables with respect to the input data, hence we will seek to retain coefficients that are ”unscaled”. Such coefficients will need to account for the relative portions of variance carried by the principal components.

Definition 4. *(Weighted Scores) Given the matrix $A \in R^{n\times m}$ with rank $r = rank(A)$ s.t. $r \leqslant min\{n, m\}$ and $k \leqslant r$. Let V_k be the matrix containing the top k right singular vectors of A, and S_k be the matrix containing the top k singular values of A. Then, the (rank-k) weighted score of the i-th column of A is defined as $\hat{w}_i^{(k)} = |\Sigma_{j=1}^k w_j e_{i,j}|$, for $i = 1, 2, ..., m$*
where:

- *$w_j = \lambda_j / \Sigma_{z=1}^r \lambda_z$, the fraction of variance carried by the j-th column in $[V_k]$, for $1 \leqslant j \leqslant k$ and,*

– *$\lambda_j = \sigma_j^2/(n-1)$ is the variance corresponding to the j^{th} singular value(σ_j), consequently to the j^{th} column of $[V_k]$, and $\lambda_1 \geq \ldots \geq \lambda_r \geq 0$*

Let us note that the weight w within the weighted score is reflected through the proportion of explained variance retained by the specific principal component. For instance if we consider the j^{th} principal direction, its weight labeled w_j is $w_j = \lambda_j / \Sigma_{z=1}^{r} \lambda_z$.
A matrix $[wV_k]$ of weighted principal directions is then constructed by multiplying each component within the retained matrix of eigenvectors V_k by its corresponding weight w_j.

$$V_k = \begin{array}{c} \begin{matrix} E_1 & E_2 & \cdots & E_k \end{matrix} \\ \begin{bmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,k} \\ e_{2,1} & e_{2,2} & \cdots & e_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ e_{m,1} & e_{m,2} & \cdots & e_{m,k} \end{bmatrix} \end{array} \qquad [wV_k] = \begin{array}{c} \begin{matrix} w_1E_1 & w_2E_2 & \cdots & w_kE_k \end{matrix} \\ \begin{bmatrix} w_1e_{1,1} & w_2e_{1,2} & \cdots & w_ke_{1,k} \\ w_1e_{2,1} & w_2e_{2,2} & \cdots & w_ke_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ w_1e_{m,1} & w_2e_{m,2} & \cdots & w_ke_{m,k} \end{bmatrix} \end{array}$$

Subsequently, the row entries of the weighted matrix $[wV_k]$ are aggregated as per line 9 of Algorithm 2 to provide the variable weights vector $\hat{w}$.

$$\hat{w} = \begin{bmatrix} |w_1e_{1,1} + w_2e_{1,2} + \cdots + w_ke_{1,k}| \\ |w_1e_{2,1} + w_2e_{2,2} + \cdots + w_ke_{2,k}| \\ \vdots \quad \vdots \quad \ddots \quad \vdots \\ |w_1e_{m,1} + w_2e_{m,2} + \cdots + w_ke_{m,k}| \end{bmatrix} = \begin{bmatrix} \hat{w_1} \\ \hat{w_2} \\ \vdots \\ \hat{w_m} \end{bmatrix}$$

The variable weights vector $\hat{w}$ entries expressed as $\hat{w_j} = |\Sigma_{z=1}^{k} w_z e_{j,z}|$ for $j = \{1, 2, ..., m\}$, are the original weights for the column-variables within the given multivariate dataset.

Deriving the Univariate Signal. Once the variable weights are uncovered, the next step consists of building a weighted matrix $[\hat{w}A]$ by factoring the original data matrix $A_{n\times m}$ and the variable weights vector $\hat{w}$. More precisely, as shown on lines 10 and 11 of Algorithm-2, each column of $A_{n\times m}$ is factored by its corresponding weight and the row entries of the weighted matrix are subsequently aggregated to form the new univariate derivation.

$$U_{n,1} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,m} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,m} \end{bmatrix} * \begin{bmatrix} \hat{w_1} \\ \hat{w_2} \\ \vdots \\ \hat{w_m} \end{bmatrix} = \begin{bmatrix} \Sigma_{v=1}^{m} a_{1v}\hat{w}_v \\ \Sigma_{v=1}^{m} a_{2v}\hat{w}_v \\ \vdots \\ \Sigma_{v=1}^{m} a_{nv}\hat{w}_v \end{bmatrix}$$

An important aspect for this representation technique is that, it uses statistics drawn from the PCA to leverage the relative importance of each variable and uncovers a univariate derivation of the time series. The new derivation takes into account the correlation between variables in the MTS dataset and, decreases redundancy and noise. The proposed representation will allow efficient UTS pattern recognition techniques to be easily extended to MTS.

5.2 Similarity Measure

We use Pearson's product-moment coefficient [21] as the measure to assess similarity between two time series. The Pearson correlation measure is known to be more robust against data that is not normalized and to respond better to baseline and scale shifts when compared to other measures [31].

Let X, Y be two normally distributed time series of equal dimension n. The Pearson correlation coefficient of X, Y denoted $\rho(X,Y)$ is a value in [-1,1] that measures of the linear dependency between X and Y, defined as follows:

$$\rho(X,Y) = \frac{\sum_{t=1}^{n}(x_t - \overline{x})(y_t - \overline{y})}{\sqrt{\sum_{t=1}^{n}(x_t - \overline{x})^2}\sqrt{\sum_{t=1}^{n}(y_t - \overline{y})^2}} \tag{1}$$

where $\overline{x}_t$ is the mean of X over n and, $\overline{y}$ is the mean of Y over n. The Pearson correlation coefficient can be approximated to the Pearson product moment, expressed as follows:

$$\rho(X,Y) = \frac{1}{n-1}\sum_{t=1}^{n}\frac{xy}{S_x S_y} \tag{2}$$

where $x = (x_t - \overline{x})\,, y = (y_t - \overline{y})$,
S_x= $[(1/n-1)\sum_{t=1}^{n} x^2]^{1/2}$, and S_y=$[(1/n-1)\sum_{t=1}^{n} y^2]^{1/2}$.

Given a user specified correlation threshold ϵ, our goal is to identify all pairs of time series whose Pearson correlation value is no less than ϵ. Algorithm 2 summarizes the steps for the pairwise correlation search from line 14 to 17.

6 Experimental Set Up and Results

To evaluate the effectiveness of our proposed technique, we implemented the code in Matlab and conducted numerous experiments on benchmark datasets, using a configured PC with Intel Quad core i7 2.00 GHz CPU, 8 GB RAM, running Windows 7.

6.1 Benchmark Datasets

The experiments were ran on benchmark datasets drawn from several widely used repositories [1, 16, 9] in the current literature. Experiments and results pertaining to three of the used benchmark datasets are reviewed in this section.

Algorithm 2 - M2U and Pairwise Correlation Search

Input: $D' = \{A^1_{n,m}, A^2_{n,m}..., A^q_{n,m}\}$ a set of normalized MTS, θ (cumulative variance explained), ϵ a user specified Pearson correlation threshold.
Output:A set C of all pairs (A^i, A^j) in D' whose correlation is not less than ϵ.

```
begin

Estimate $k_{max}$ using Algorithm 1.
        $k \leftarrow k_{max}$
for $i \leftarrow 1$ to $q$ do
    STEP1: Reduce MTS $A^i \in D'$ to UTS $U^i$, add it to $D^U$
        $A \leftarrow$ the $i^{th}$MTS of rank r, in D', $A^i_{n,m}$
        Compute the Singular Value Decomposition
        $[U, S, V^T] \leftarrow SVD(A)$
        Retain a matrix of $k$ eigenvectors
        $M \leftarrow V_k$
        Build the weighted matrix $[wV_k]$
        For z ← 1 to k
            $w_z \leftarrow \lambda_z / \Sigma^r_{z=1} \lambda_z$
            $[wV_k]_{*,z} \leftarrow w_z * [M]_{*,z}$
        end for
        Compute the weighted score for each variable
            $\hat{w}^{(k)}_j \leftarrow |\Sigma^k_{z=1} w_z e_{j,z}|$, for all j = {1, 2, ..., m}.
        Build the weighted matrix $[\hat{w}A]$
        For v ← 1 to m
            $\hat{w}_v \leftarrow \hat{w}^{(k)}_v$
            $[\hat{w}A]_{*,v} \leftarrow [A]_{*,v} * \hat{w}_v$
        end for
        Uncover row entries for the new univariate signal $U_{n,1}$
            $[U_{n,1}]_i \leftarrow \Sigma^m_{v=1} a_{i,v} \hat{w}_v$, for i = {1, 2, ..., n}
        add $[U_{n,1}]$ to $D^U$
end for
STEP2: Uncover correlated pairs
    For all $(U^i, U^j) \in D^U$
        Compute their pairwise Pearson correlations
        If ( $|\rho(U^i, U^j)| \geqslant \epsilon$ ) then add $(A^i, A^j)$ to $C$

end
```

The Australian language sign dataset(AUSLAN) [12] was gathered through two gloves, with 22 sensors while native AUSLAN speakers signed. The dataset contains 95 signs having 27 examples each, hence a total of 2565 of signs gathered. This dataset is well used in similarity search problems due to its complexity.

The INRIA Holidays images dataset (INRIA HID) [9] is a collection of images that have served in testing the robustness to various transformations: rotations, viewpoint and illumination changes, blurring, etc. The dataset contains 500 high resolution image groups representing a large variety of scene types to incorporate diversity in representation.

The Transient classification benchmark dataset (Trace) [25] was gathered for power plant diagnostics. The dataset has 5 variables (4 process variables and a class label) and 16 operating states. The class label is set to 0 until the transient occurs, at which time it is set to 1. The part of the data that is of interest for us is the subset where the transient occurs.

6.2 Evaluation and Results

We designed experiments to assess the performance of the proposed technique. In this section, we compare our performance against those from primarily five other techniques: the Correlation Based Dynamical Time Warping (CBDTW) [2], the 2-D correlation measure for matrices(see section 6.2)(Corr2), the Dynamical Time Warping(DTW), Eros [29], Euclidien Distance(ED).

The recall-precision ratios recorded for all techniques on the AUSLAN and TRACE datasets are shown on Fig. 1 and Fig. 2 respectively. On both datasets, we can see that the Euclidien Distance(ED), Dynamical Time Warping(DTW) perform worst than the remaining techniques. This may be due to the fact that, both techniques do not take into account the existing correlations between the variables of the MTS while the remaining four techniques do. Our technique outperforms the remaining techniques on both datasets. In another set of experiments, we further assess how using the proposed univariate representations

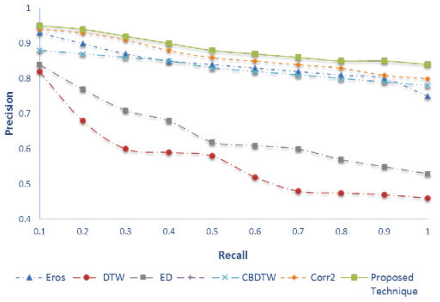

Fig. 1. Recall-Precision on AUSLAN

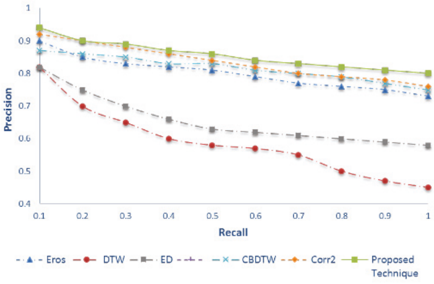

Fig. 2. Recall-Precision on TRACE

compares to the case where the original matrices are used to find pairwise correlations within a set of MTS. Our results confirm that our technique yields improved similarity search accuracy. To illustrate with an example from the INRIA Holidays images dataset, let us consider the six images on the left side of Fig. 3, with three scenes taken at different points in time.

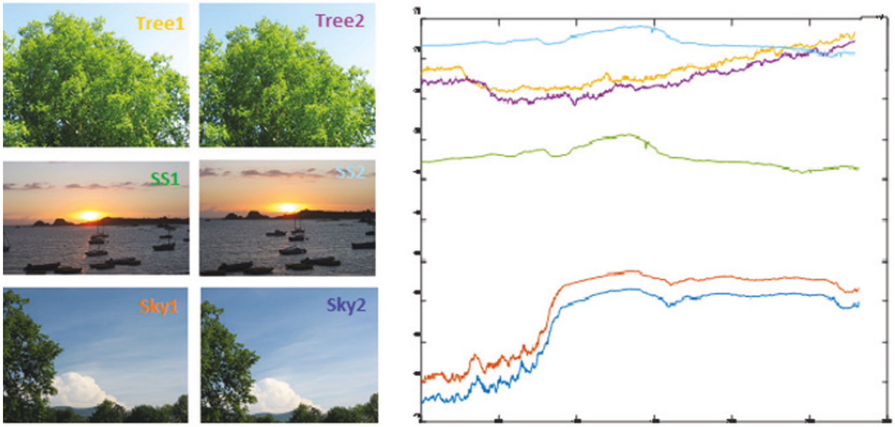

Fig. 3. Left - Six images from the INRIA HID of three scenes taken at different points in time, identified as closest matches. Right - Univariate signals of the six images after M2U transformation. The image name is color-coded with its corresponding univariate signal.

For the purpose of the experiment, the images were converted to the grayscale intensity images, then subsequently to double precision to transform the truecolor image RGB to 2-dimensional matrices. Each image was then represented by a 2816×2112 matrix.

Using our proposed transformation technique M2U, each matrix is transformed into a univariate signal represented on the right of Fig. 3. The color of each univariate signal (Fig. 3 right) matches the color of the text on its corresponding image to the left side (Fig. 3 left). We can see that similar images generated similar univariate signals. Furthermore, using our technique, the Pearson correlation coefficients post transformation are:

ρ(Tree1,Tree2)=0.9661,
ρ(SS1,SS2)=0.9413,
ρ(Sky1,Sky2)=0.9982.

To uncover the correlation coefficient that would result from using the original matrices, without transformation, we use the 2-D correlation coefficient *Corr2*, framed as:

$$Corr2(A^i, A^j) = \frac{\Sigma_n \Sigma_m (A^i_{mn} - \bar{A}^i)(A^j_{mn} - \bar{A}^j)}{\sqrt{(\Sigma_n \Sigma_m (A^i_{mn} - \bar{A}^i)^2)(A^j_{mn} - \bar{A}^j)^2}}$$

where $\bar{A}^i = mean2(A^i)$ and $\bar{A}^j = mean2(A^j)$.

For this set of experiments on the full image matrices, the Pearson correlation coefficients are such as:

Corr2(Tree1,Tree2)=0.6261,
Corr2(SS1,SS2)=0.7594,
Corr2(Sky1,Sky2)=0.8027.

Let us consider the case where we looked to retrieve all similar images, with a correlation coefficient greater than a correlation threshold $\epsilon = 0.7$, the images for Tree1 and Tree2 would not have been retrieved as pairwise correlated if the full matrix is used, while it would be identified if the Pearson correlation is applied to the univariate derivation obtained from our technique (M2U).

Transforming the MTS to a univariate signal yields improved similarity search accuracy. When the matrix goes through a PCA transformation, in addition to reducing the dimensionality, the transformation decreases redundancy and noise, highlights relationships between the different variables and reveals patterns by compressing the data while expressing it in such a way that highlights their similarity and dissimilarity. In addition, since we are not discarding any of the relevant principal components, but rather re-combing them, we preserve much of the relevant and needed information from the data.

7 Conclusion

We propose a novel technique for multivariate time series representation, analysis and search. The technique relies on dimensionality reduction and correlation analysis to uncover similar multivariate time series. It uses statistics drawn from the Principal Component Analysis to find a unique derivation of the MTS into a univariate time series prior to seeking correlations. Our experiment results indicate increased accuracy and efficiency when compared to major existing techniques. The proposed representation will allow efficient techniques for univariate time series to be easily extended to multivariate time series. We are currently working on extending the proposed technique to application frameworks for streaming time series.

Acknowledgments: This work was partially supported by Natural Sciences and Engineering Research Council (NSERC) of Canada and Concordia University.

References

1. A. Asuncion and D. Newman. Uci machine learning repository, 2007.
2. Z. Bankó and J. Abonyi. Correlation based dynamic time warping of multivariate time series. *Expert Systems with Applications*, 39(17):12814–12823, 2012.
3. E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250. ACM,2001.
4. A. Camerra, T. Palpanas, J. Shieh, and E. Keogh. isax 2.0: Indexing and mining one billion time series. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 58–67. IEEE, 2010.
5. B. A. Draper, K. Baek, M. S. Bartlett, and J. R. Beveridge. Recognizing faces with pca and ica. *Computer vision and image understanding*, 91(1):115–137, 2003.
6. B. Esmael, A. Arnaout, R. K. Fruhwirth, and G. Thonhauser. Multivariate time series classification by combining trend-based and value-based approximations. In *Computational Science and Its Applications–ICCSA 2012*, pages 392–403. Springer, 2012.
7. D. Fradkin and D. Madigan. Experiments with random projections for machine learning. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 517–522. ACM, 2003.
8. A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4):411–430, 2000.
9. H. Jegou, M. Douze, and C. Schmid. Inria holidays dataset, 2008.
10. W. B. Johnson and J. Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.
11. I. Jolliffe. *Principal component analysis.* Wiley Online Library.
12. M. W. Kadous. *Temporal classification: Extending the classification paradigm to multivariate time series.* PhD thesis, The University of New South Wales, 2002.
13. T. Kahveci, A. Singh, and A. Gurel. Similarity searching for multi-attribute sequences. In *Scientific and Statistical Database Management, 2002. Proceedings. 14th International Conference on*, pages 175–184. IEEE, 2002.
14. A. Kane and N. Shiri. Selecting the top-k discriminative features using principal component analysis. In *Data Mining Workshops (ICDMW), 2016 IEEE 16th International Conference on*, pages 639–646. IEEE, 2016.
15. L. Karamitopoulos, G. Evangelidis, and D. Dervos. Pca-based time series similarity search. In *Data Mining*, pages 255–276. Springer, 2010.
16. E. Keogh. Ucr time series archive www. cs. ucr. edu/˜ eamonn, 2006.
17. E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and information Systems*, 3(3):263–286, 2001.
18. J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 2–11. ACM, 2003.
19. M. Moinester and R. Gottfriedb. Sample size estimation for correlations with pre-specified confidence interval.
20. A. Mueen, S. Nath, and J. Liu. Fast approximate correlation for massive time-series data. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 171–182. ACM, 2010.

21. K. Pearson. Mathematical contributions to the theory of evolution. xix. second supplement to a memoir on skew variation. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, pages 429–457, 1916.
22. Quandl *url http://www.quandl.com/help/api*.
23. T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. pages 262–270, 2012.
24. C. Ratanamahatana, E. Keogh, A. J. Bagnall, and S. Lonardi. A novel bit level time series representation with implication of similarity search and clustering. In *Advances in knowledge discovery and data mining*, pages 771–777. Springer, 2005.
25. D. Roverso. Plant diagnostics by transient classification: The aladdin approach. *International Journal of Intelligent Systems*, 17(8):767–790, 2002.
26. J. Shieh and E. Keogh. isax: disk-aware mining and indexing of massive time series datasets. *Data Mining and Knowledge Discovery*, 19(1):24–57, 2009.
27. Y. Tanaka, K. Iwamoto, and K. Uehara. Discovery of time-series motif from multidimensional data based on mdl principle. *Machine Learning*, 58(2-3):269–300, 2005.
28. K. Yang and C. Shahabi. A pca-based similarity measure for multivariate time series. In *Proceedings of the 2nd ACM international workshop on Multimedia databases*, pages 65–74. ACM, 2004.
29. K. Yang, H. Yoon, and C. Shahabi. A supervised feature subset selection technique for multivariate time series.yang2004pca 2005.
30. B.-K. Yi and C. Faloutsos. Fast time sequence indexing for arbitrary lp norms. VLDB, 2000.
31. Y. Zhu. *High performance data mining in time series: techniques and case studies*. PhD thesis, New York University, 2004.