Knowledge-Based Systems 21 (2008) 535–539

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

# Classification of multivariate time series using two-dimensional singular value decomposition

Xiaoqing Weng [a,b,*], Junyi Shen [a]

[a] Institute of Computer Software, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China
[b] Computer Center of Hebei University of Economics and Trade, Shijiazhuang, China

ARTICLE INFO

*Article history:*
Received 18 April 2007
Accepted 21 March 2008
Available online 29 March 2008

*Keywords:*
Two-dimensional singular value decomposition
Multivariate time series
Classification

ABSTRACT

Multivariate time series (MTS) are used in very broad areas such as multimedia, medicine, finance and speech recognition. A new approach for MTS classification using two-dimensional singular value decomposition (2dSVD) is proposed. 2dSVD is an extension of standard SVD, it captures explicitly the two-dimensional nature of MTS samples. The eigenvectors of row–row and column–column covariance matrices of MTS samples are computed for feature extraction. After the feature matrix is obtained for each MTS sample, one-nearest-neighbor classifier is used for MTS classification. Experimental results performed on five real-world datasets demonstrate the effectiveness of our proposed approach.



## 1. Introduction

Multivariate time series (MTS) are used in very broad areas such as multimedia, medicine, finance and speech recognition. MTS classification is an important problem in time series data mining. MTS is a series of observations, $\mathbf{x}_i(t)[i = 1, 2, \ldots, n; t = 1, 2, \ldots, m]$, where $m$ is the number of observations and $n$ is the number of variables ($n$ is greater than, or equal to 2) [1]. MTS sample is stored in an $m \times n$ matrix.

Given a MTS dataset $\{T_i\}_{i=1}^{M+N}$, where each MTS sample $T_i \in R^{m\times n}$. First $M$ MTS samples $\{\mathbf{T}_i \mid 1 \leqslant i \leqslant M\}$ are labeled. The task of MTS classification is to estimate the labels of unlabeled $N$ MTS samples $\{\mathbf{T}_i \mid M+1 \leqslant i \leqslant M+N\}$. MTS classification is difficult for traditional machine learning algorithms mainly because of dozens of variables and different lengths of MTS samples. MTS sample is in fact one kind of two-dimensional data. Several approaches have been proposed for MTS classification [2–9]. However, these approaches for MTS classification do not consider explicitly the two-dimensional nature of MTS samples.

In this paper, a new approach for MTS classification using two-dimensional singular value decomposition (2dSVD) [10] is proposed. 2dSVD is an extension of standard SVD. 2dSVD captures explicitly the two-dimensional nature of the two-dimensional objects, such as two-dimensional images, two-dimensional weather maps, etc. To the best of our knowledge, 2dSVD has not been previously investigated for MTS classification. In our approach, the eigenvectors of row–row and column–column covariance matrices of MTS samples are computed for feature extraction. After the feature matrix is obtained for each MTS sample, one-nearest-neighbor classifier (1NN) is used for MTS classification. Li et al. [4,6] proposed two feature vector selection approaches from MTS sample using standard SVD (1dSVD). Our proposed approach is compared with Li's two approaches. Experimental results performed on five real-world datasets demonstrate the effectiveness of our approach.

The remainder of this paper is organized as follows. Section 2 gives a brief review of related work and background. Section 3 proposes a new approach for MTS classification using 2dSVD. Section 4 experimentally demonstrates the effectiveness of our proposed approach. Conclusion is presented in Section 5.

## 2. Related works and background

### 2.1. Related works

There are some works that also address MTS classification problem.

Li et al. [4,6] proposed two different feature vector selection approaches for MTS classification by using stand SVD (1dSVD). Let $\mathbf{Q}$ denote the MTS sample, $\mathbf{Q}$ is stored in an $m \times n$ matrix, where $m$ is the number of observations and $n$ is the number of variables, $\mathbf{u}_1$ is the first singular vector of $\mathbf{Q}$, $\mathbf{u}_2$ is the second singular vector of $\mathbf{Q}$, $\sigma$ is the vector of the singular values of $\mathbf{Q}^{\mathrm{T}}\mathbf{Q}$, and $\sigma_{\mathrm{std}} = \sigma / \mid \sigma \mid$ is the normalized singular value vector. The first approach (Li's first) considers the first singular vector and the normalized singular values, i.e. the first singular vector $\mathbf{u}_1$ concatenated by the normalized sin-

* Corresponding author. Address: Institute of Computer Software, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China.

*E-mail addresses:* xqweng@mail.xjtu.edu.cn (X. Weng), jyshen@mail.xjtu.edu.cn (J. Shen).

0950-7051/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2008.03.014

gular value vector $\sigma_{std}$. The second approach (Li's second) takes into account the first two dominating singular vectors weighted by their associated singular values, i.e. the weighted first singular vector $w_1\mathbf{u}_1$ concatenated by the weighted second singular vector $w_2\mathbf{u}_2$, $w_i = \sigma_{std}(i)/\sum_{j=1}^{n}\sigma_{std}(j)$. We refer to the two different feature vector selection approaches as Li's first and Li's second approach hereafter.

The hypothesis test and clustering have been used to compare and classify stationary MTS [7]. *p*-Value of the test of hypothesis is defined for measuring the similarity of two stationary MTS. Hierarchical clustering is used to classify stationary MTS. Geurts et al. [8] designed a general approach for MTS classification, which classifies subsequences instead of the whole MTS sample. A number of randomly selected subsequences are extracted from the MTS sample, and classifiers are built on these subsequences, MTS samples can be then classified by votes on its subsequences classifiers. Rodriguez et al. [2] proposed to select literals from MTS samples with boosting and using these literals with SVM. Kadous et al. [5] proposed an approach for MTS classification by using metafeature. Recurring substructures (termed metafeatures) in MTS sample are extracted to construct a set of attributes, MTS samples can be then classified by using standard learners. Hayashi et al. [3] proposed an approach to embed MTS samples in a vector space and to classify them in the embedded space. Yang et al. [9] proposed a new feature subset selection method for MTS classification, based on common principal component analysis.

MTS sample is in fact one kind of two-dimensional data. However, the above approaches for MTS classification do not consider explicitly the two-dimensional nature of MTS samples.

With respect to 2dSVD, one of the most related works is two-dimensional principal component analysis (2DPCA) [14]. Wang et al. [15] have shown that 2DPCA is equivalent to a special case of block-based PCA. 2DPCA can be thought as a one-sided low-rank approximation with its optimal solution given by 2dSVD [10].

### 2.2. Background

2dSVD [10] is an extension of standard SVD. 2dSVD is based on two-dimensional matrices rather than one-dimensional vectors. MTS row–row and column–column covariance matrices are constructed directly using the original MTS samples and their eigenvectors are computed for MTS feature extraction. In low rank approximation, 2dSVD captures explicitly the two-dimensional nature of the two-dimensional objects, such as two-dimensional images, two-dimensional weather maps. We give a brief description of 2dSVD as follows (see [10] for details).

Given a MTS dataset $\{T_i\}_{i=1}^{c}$, where each MTS sample $T_i \in R^{m\times n}$. Define the averaged row–row covariance matrix $\mathbf{F}$ and column–column covariance matrix $\mathbf{G}$

$$\mathbf{F} = \frac{1}{c}\sum_{i=1}^{c}(T_i - \overline{T})(T_i - \overline{T})^{\mathrm{T}}, \tag{1}$$

$$\mathbf{G} = \frac{1}{c}\sum_{i=1}^{c}(T_i - \overline{T})^{\mathrm{T}}(T_i - \overline{T}), \tag{2}$$

where $\overline{T} = \sum_i T_i/c$. Let $\mathbf{U}_r$ contains $r$ principal eigenvectors of $\mathbf{F}$ and $\mathbf{V}_s$ contains $s$ principal eigenvectors of $\mathbf{G}$, $U_r \equiv (u_1,\ldots,u_r)$ and $V_s \equiv (v_1,\ldots,v_s)$. The 2dSVD of $\{T_i\}_{i=1}^{c}$ is $(U_r, V_s, \{M_i\}_{i=1}^{c})$, where $M_i = U_r^{\mathrm{T}}T_iV_s$, $U_r \in R^{m\times r}$, $V_s \in R^{n\times s}$, $M_i \in R^{r\times s}$. $\mathbf{M}_i$ is the feature matrix of MTS sample $\mathbf{T}_i$. A feature matrix is obtained for each MTS sample by using 2dSVD. Let $\mathbf{M}_i = [Y_1^i, Y_2^i, \ldots, Y_s^i]$ denote the feature matrix of MTS sample $\mathbf{T}_i$, $i = 1,2$. The distance between two feature matrices $d(\mathbf{M}_1, \mathbf{M}_2)$ is defined as

$$d(\mathbf{M}_1,\mathbf{M}_2) = \sum_{k=1}^{s}\|Y_k^1 - Y_k^2\| \tag{3}$$

where $\|\bullet\|$ means $L_2$ norm.

**Definition 1.** Given a MTS sample $T \in R^{m\times n}$ and its reconstructed MTS sample $\overline{T} = U_rM_iV_s^{\mathrm{T}}$, the squared errors between $\mathbf{T}$ and $\overline{T}$ is defined as [2]:

$$\mathrm{SE} = \sum_{i=1}^{m}\sum_{j=1}^{n}(T(i,j) - \overline{T(i,j)})^2 \tag{4}$$

## 3. MTS classification using 2dSVD

MTS classification using 2dSVD has three steps. First, we calculate the transformations $\mathbf{U}_k$ and $\mathbf{V}_s$ from the training MTS samples and obtain a feature matrix for each training MTS sample; then the feature matrix of the each testing MTS sample is calculated; finally, the MTS samples in testing dataset can be identified by one-nearest-neighbor classifier with distance (3). The algorithmic procedure is stated below.

**Algorithm 1.** Two_SVD_1NN(TRAIN, TEST)

*Input:* training dataset, TRAIN, testing dataset, TEST.
*Output:* error rate of classification.

1. Compute row–row covariance matrix, $\mathbf{F}$.
2. $\mathbf{U}_r \leftarrow r$ principal eigenvectors of $\mathbf{F}$.
3. Compute column–column covariance matrix, $\mathbf{G}$.
4. $\mathbf{V}_s \leftarrow s$ principal eigenvectors of $\mathbf{G}$.
5. $M_i = U_r^{\mathrm{T}} * \mathrm{TRAIN}(i) * V_s$ for $\forall\ \mathrm{TRAIN}(i) \in \mathrm{TRAIN}$.
6. $N_i = U_r^{\mathrm{T}} * \mathrm{TEST}(i) * V_s$ for $\forall \mathrm{TEST}(i) \in \mathrm{TRAIN}$.
7. Perform one-nearest-neighbor classifier on embedded results $\{\mathbf{M}_i, \mathbf{N}_i\}$.
8. Return the error rate of classification.

Let $M$ denote the number of MTS samples in training dataset, $N$ denote the number of MTS samples in testing dataset. row–row covariance matrix $\mathbf{F}$ is $m \times m$ matrix, column–column covariance matrix $\mathbf{G}$ is $n \times n$ matrix.

The time complexity from Step (1) to Step (4) is $O(m^3 + n^3)$, since the eigenvectors can be computed by performing SVD on the covariance matrix and the complexity of SVD for $n \times n$ matrix is $O(n^3)$ [1]. The complexity of extracting feature matrices from MTS samples in training dataset and in testing dataset (Step (5) and Step (6)) is $O((M+N) * r * s)$, where $r$ is the number of the principal eigenvectors of row–row covariance matrix, $s$ is the number of the principal eigenvectors of column–column covariance matrix. The complexity of performing one-nearest-neighbor classifier (Step (7)) is $O(M * N)$. In general, $n$, $r$ and $s$ are much smaller than the length of MTS sample or the number of observations, $m$. Therefore, the complexity of our algorithm is determined by the length of MTS sample $m$.

The algorithm assumes that parameters $r$ and $s$ is known a priori. However, this may not be a reasonable assumption in many applications.

Since the error rate of our algorithm decreases as the number of principal eigenvectors of column–column covariance matrix $s$ increases (see Section 4.3 for details), therefore, $s$ is chosen in this work such that $s$ eigenvectors can describe at least 98% of the total column–column variations in MTS dataset.

We propose a method that estimates the parameter $r$ by observing the behavior of the sum of squared errors of all training samples SSE($r$), for different values of $r$.

$$\mathrm{SSE}(r) = \sum_{T\in\mathrm{TRAIN}}\sum_{i=1}^{m}\sum_{j=1}^{n}(T(i,j) - \overline{T_r(i,j)})^2, \quad r = 1,2,\ldots,m \tag{5}$$

The percentage change in SSE($r$) is calculated as:

$$\mathrm{dSSE}(r) = \frac{|\,\mathrm{SSE}(r+1) - \mathrm{SSE}(r)\,|}{\mathrm{SSE}(r)} \times 100\%, \quad r = 1, 2, \ldots, m-1 \qquad (6)$$

It is convenient to repeat our algorithm for a fixed parameter $s$ and then observe the behavior of SSE($r$) in order to estimate the appropriate parameter $r$, the value of $r$ is increased from one to $m-1$, where $m$ is the length of MTS samples. Generally, SSE($r$) decreases with increasing $r$, however, the appropriate parameter $r$ can be estimated if the value of SSE($r$) varies significantly between consecutive values of $r$. Let $r^* = \arg\max(\mathrm{dSSE}(r))$, the appropriate parameter $r$ is defined as the $r^*$ or $r^*+1$. If $r^* = 2$, the appropriate parameter $r$ may be 1.

## 4. Experiments

In experiments, we adopt 2dSVD and Li's two approaches [4,6] for feature extraction, one-nearest-neighbor (1NN) classifier with Euclidean distance for classification, note that distance (3) is used to calculate the distance between two feature matrices extracted by using 2dSVD.

In principle, any learner (SVM, *k*NN, Bayesian, etc.) could be used in this work. However, it has been shown that 1NN with Euclidean distance is very hard to beat [16,17,20]. The advantage of 1NN is that it does not need parameter. In this work, we apply 1NN classifier for its simplicity.

Several experiments are carried out to show the effectiveness of our proposed approach (2dSVD + 1NN) for MTS classification. Our proposed approach is compared with Li's two approaches +1NN. The accuracy of classification is measured by the classification error rate.

The experiments are implemented in Matlab6.5 and performed on a Pentium IV 1.8 GHz machine with 256 M of main memory.

### *4.1. Datasets*

The experiments are carried out on five real-world datasets, i.e. AUSLAN (Australian Sign Language) dataset [11], Japanese Vowels dataset [11], BCI (Brain Computer Interface) dataset [12,13], Wafer dataset and ECG dataset [18,19], which are all labeled dataset whose labels are given.

AUSLAN dataset contains 95 signs, 27 samples per sign, and there are 2565 samples in AUSLAN dataset. Samples of each sign were captured from a native AUSLAN speaker using 22 sensors on the CyberGlove. Each sample can be regarded as a MTS sample with 22 variables. The average length of each sample is around 60.

We use 25 signs (hence, 675 samples in total) in our experiments. The 25 signs are alive, all, boy, building, buy, cold, come, computer, cost, crazy, danger, deaf, different, girl, glove, go, God, joke, juice, man, where, which, yes, you and zero. Twenty seven samples per sign can be regarded as one class. Random 15 samples per sign are taken with labels to form the training samples, the remaining 12 samples per sign are considered to be the testing samples.

Japanese Vowels dataset contains 640 time series of 12 LPC cepstrum coefficients (i.e. MTS sample with 12 variables) taken from nine male speakers. The length of MTS sample is between 7 and 29. The task is to distinguish nine male speakers by their utterances of two Japanese Vowels/ae/. Speakers 1–9 have the corresponding number of samples: 61, 65, 118, 74, 59, 54, 70, 80, 59. Thus, speakers 1 has 61 samples (61 utterances of/ae/), speaker 2 has 65 samples (65 utterances of/ae/), and so on. Samples per male can be regarded as one class. Random 30 samples per speaker are taken with labels to form the training samples, the remaining samples per speaker are considered to be the testing samples.

BCI dataset contains 416 samples, each sample contains 28 EEG channels (i.e. MTS sample with 28 variables), 208 samples are for upcoming left hand movements and 208 samples are for upcoming right hand movements. Class label '1' is used for left, class labels '2' is used for right. The length of each MTS sample is 500. Three hundred and sixteen samples are used for training and other 100 samples for testing.

The Wafer dataset uses six vacuum-chamber sensors to collect data while monitoring an operational semiconductor fabrication plant. Each wafer is described by a MTS sample with six variables and assigned classification of normal or abnormal. The dataset we use contains 327 MTS samples, among which 200 samples are normal and 127 samples are abnormal. The length of MTS sample is between 104 and 198. One hundred and eighty samples are used for training and other 147 samples for testing.

The ECG dataset uses two electrodes to collect data during one heartbeat. Each heartbeat is described by a MTS sample with two variables and assigned classification of normal or abnormal. Abnormal heartbeats are representative of a cardiac pathology known as supraventricular premature beat. The ECG dataset contains 200 MTS samples, among which 133 samples are normal and 67 samples are abnormal. The length of MTS sample is between 39 and 152. One hundred and seven samples are used for training and other 93 samples for testing. Table 1 shows the summary of the datasets used in the experiments.

The MTS samples in each dataset are of different lengths. 2dSVD is available only when MTS samples are of the same length. We truncate these MTS samples to the shortest length of the MTS sample in the dataset.

### *4.2. Performance comparison*

Table 2 shows the classification error rates obtained with 2dSVD and with 1dSVD (Li's first method and Li's second method) on five real-world datasets. It is found that 2dSVD has better results than 1dSVD on four of the five datasets, the exception is Japanese Vowels dataset. the error rate for 2dSVD achieves the best result with $r = 1$ and $s = 15$ for AUSLAN, $r = 3$ and $s = 7$ for BCI, $r = 1$ and $s = 2$ for ECG, $r = 8$ and $s = 3$ for Wafer. The error rates for AUSLAN, BCI, ECG and Wafer are 0.05, 0.42, 0.2688 and 0.0544, respectively.

2dSVD does not work well on Japanese Vowels dataset. This can be explained by that all MTS samples in this dataset are truncated

**Table 1**
Summary of datasets used in the experiments

| | AUSLAN | Japanese Vowels | BCI | Wafer | ECG |
|---|---|---|---|---|---|
| Number of variables | 22 | 12 | 28 | 6 | 2 |
| Max length | 95 | 29 | 500 | 198 | 152 |
| Min length | 47 | 7 | 500 | 104 | 39 |
| Number of labels | 25 | 9 | 2 | 2 | 2 |
| Total number of MTS samples | 675 | 640 | 416 | 327 | 200 |

**Table 2**
The error rates on five datasets

| Dataset | Li's first approach | Li's second approach | 2dSVD | Parameters |
|---|---|---|---|---|
| AUSLAN | 0.11 | 0.12 | 0.05 | $r = 1$, $s = 15$ |
| Japanese Vowels | 0.0784 | 0.0703 | 0.0946 | $r = 3, s = 9$ |
| BCI | 0.55 | 0.54 | 0.42 | $r = 3, s = 7$ |
| ECG | 0.4624 | 0.3978 | 0.2688 | $r = 1, s = 2$ |
| Wafer | 0.0816 | 0.1088 | 0.0544 | $r = 8, s = 3$ |

to the shortest length, the shortest length is only 7, however, the longest length is 29, large amount data is lost.

### 4.3. Influence of parameters on performance

In this section, we evaluate the influence of the parameters $r$ (i.e. the number of principal eigenvectors of row–row covariance matrix) and $s$ (i.e. the number of principal eigenvectors of column–column covariance matrix) on performance of our approach.

For AUSLAN dataset, the eight principal eigenvectors of row–row covariance matrix (i.e. $r = 8$) can describe at least 98% of the total row–row variations, and the fifteen principal eigenvectors of column–column covariance matrix (i.e. $s = 15$) can describe at least 98% of the total column–column variations. Fig. 1 shows the plot of error rate versus the parameter $r$ when $s = 15$. Fig. 2 shows the plot of error rate versus the parameter $s$ when $r = 8$.

For Wafer dataset, the twenty-eight principal eigenvectors of row–row covariance matrix (i.e. $r = 28$) can describe at least 98% of the total row–row variations, and the three principal eigenvectors of column–column covariance matrix (i.e. $s = 3$) can describe at least 98% of the total column–column variations. Fig. 3 shows the plot of error rate versus the parameter $r$ when $s = 3$.

For Japanese Vowels dataset, the three principal eigenvectors of row–row covariance matrix (i.e. $r = 3$) can describe at least 98% of the total row–row variations, and the nine principal eigenvectors of column–column covariance matrix (i.e. $s = 9$) can describe at least 98% of the total column–column variations. Fig. 4 shows the plot of error rate versus the parameter $s$ when $r = 3$.

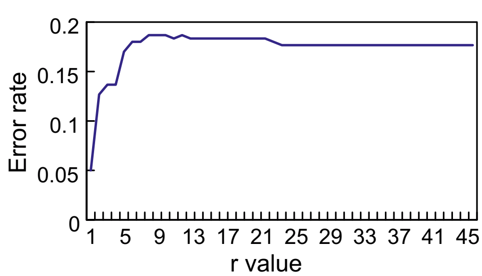


**Fig. 1.** Error rate vs. value $r$ on AUSLAN dataset.

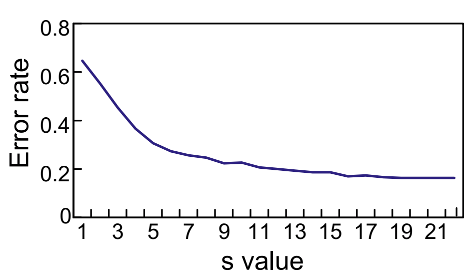


**Fig. 2.** Error rate vs. value $s$ on AUSLAN dataset.

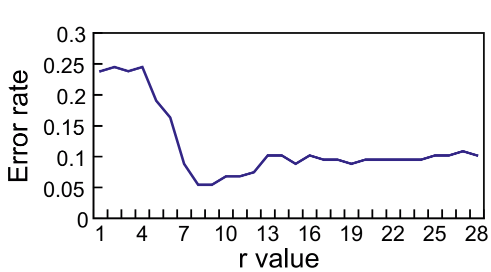


**Fig. 3.** Error rate vs. value $r$ on Wafer dataset.

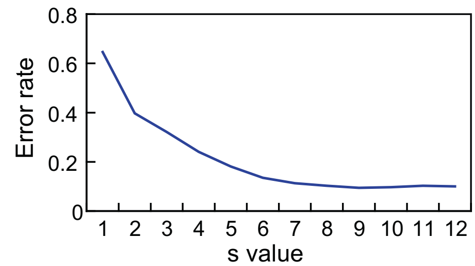


**Fig. 4.** Error rate vs. value $s$ on Japanese Vowels dataset.

As can be seen from Figs. 2 and 4, the error rate achieves the largest value when $s = 1$, and decreases as parameter $s$ increases. Therefore, larger value of $s$ should be chosen for reducing the error rate. In this work, $s$ is chosen such that $s$ eigenvectors can describe at least 98% of the total column–column variations in MTS dataset.

As observed in Fig. 1, the error rate achieves the best result with $r = 1$, the error rate is 0.05. The error rate would not reduce by increasing value $r$ since the error rate increases as parameter $r$ increases. For Wafer dataset, it is found from Fig. 3 that the error rate of our approach decreases fast as parameter $r$ increases, and achieves the best result with $r = 8$. Therefore, We can reduce the error rate by choosing appropriate parameter $r$.

For Wafer dataset, Fig. 5 shows the plot of SSE versus the parameter $r$ when $s = 3$. It is found that SSE (i.e. the sum of squared errors of all training samples) decreases fast as parameter $r$ increases. Fig. 6 shows the plot of dSSE versus the parameter $r$, dSSE achieves the largest value with $r^* = 7$, dSSE(7) = 53.52. As can be seen from Fig. 3, the error rate achieves the best result with $r = r^* + 1 = 7 + 1 = 8$, the appropriate parameter $r$ for Wafer dataset is 8.

### 4.4. Experimental results on the extended datasets

It could be argued that truncating MTS samples to the length of the shortest MTS sample in dataset is not adequate for using

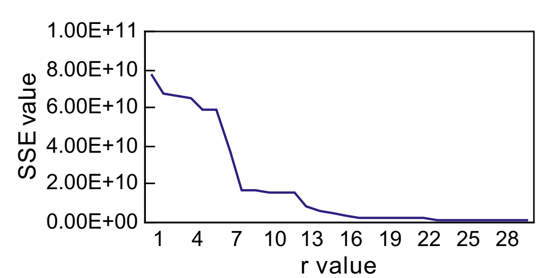


**Fig. 5.** SSE vs. value $r$ on Wafer dataset.

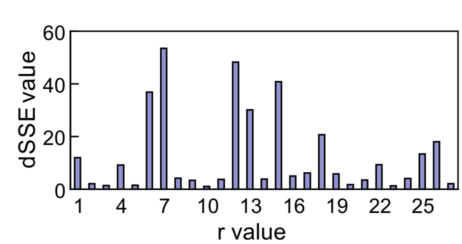


**Fig. 6.** dSSE vs. value $r$ on Wafer dataset.

**Table 3**
The error rates on the extended datasets

| Dataset | Li's first approach | Li's second approach | 2dSVD | Parameters |
|---|---|---|---|---|
| AUSLAN | 0.1033 | 0.1033 | 0.0667 | $r = 1$, $s = 15$ |
| Japanese Vowels | 0.0703 | 0.0595 | 0.0459 | $r = 3$, $s = 9$ |
| ECG | 0.4731 | 0.4516 | 0.2903 | $r = 12$, $s = 2$ |
| Wafer | 0.0272 | 0.0340 | 0.0136 | $r = 24$, $s = 6$ |

2dSVD. Another alternative is considered. For each dataset, the MTS samples are extended to the length of the longest MTS sample in the dataset [2]. Some of the values in MTS sample are duplicated in order to extend a MTS sample. For instance, if we want to extend a MTS sample of length 60 to a length of 75, one of each four values would be duplicated. In this way, all the values in original MTS sample are in the extended MTS sample. Table 3 shows the results on the extended datasets. It is found that 2dSVD has better results than 1dSVD on the four extended datasets by choosing appropriate parameters $r$ and $s$.

## 5. Conclusion

In this paper, a new approach for MTS classification using 2dSVD has been presented. The eigenvectors of row–row and column–column covariance matrices of MTS samples are computed for feature extraction. After the feature matrix is obtained for each MTS sample, one-nearest-neighbor classifier is used for MTS classification. By choosing appropriate parameters $r$ and $s$, our approach outperforms 1dSVD on five real-world datasets.

We intend to extend this work in two directions. First, we plan to use feature matrices extracted from MTS samples to MTS clustering. Clustering on these feature matrices instead of clustering on MTS samples directly may get better results. Second, we plan to propose a 2dSVD-based index structure for efficient retrieval of MTS samples. The 2dSVD-based index structure may get better efficiency since 2dSVD captures the two-dimensional nature of MTS samples.

## Acknowledgements

We thank the maintainers of the UCI KDD Repository [11] and the donors of the different datasets [13,19]. This work was supported by the National Science Foundation of China under the Grant No. 60173058.

## References

[1] K. Yang, C. Shahabi, A PCA-based similarity measure for multivariate time series, in: Proceedings of the Second ACM International Workshop on Multimedia Databases, 2004, pp. 65–74.
[2] J.J. Rodriguez, C.J. Alonso, J.A. Maestro, Support vector machines of interval-based features for time series classification, Knowledge-Based Systems 18 (2005) 171–178.
[3] A. Hayashi, Y. Mizuhara, N. Suematsu, Embedding time series data for classification, in: Proceedings of the MLDM 2005, LNAI, vol. 3587, 2005, pp. 356–365.
[4] C. Li, L. Khan, B. Prabhakaran, Real-time classification of variable length multi-attribute motion data, International Journal of Knowledge and Information Systems 10 (2) (2006) 163–183.
[5] M.W. Kadous, Classification of multivariate time series and structured data using constructive induction, Machine Learning 58 (2005) 179–216.
[6] C. Li, L. Khan, B. Prabhakaran, Feature selection for classification of variable length multi-attribute motions, in: V.A. Petrushin, L. Khan (Eds.), Multimedia Data Mining and Knowledge Discovery, Springer-Verlag, 2007 (Chapter 7). Available from: <http://www.utdallas.edu/~chuanjun/publication.htm/>.
[7] E.A. Maharaj, Comparison and classification of stationary multivariate time series, Pattern Recognition 32 (7) (1999) 1129–1138.
[8] P. Geurts, L. Wehenkel, Segment and combine approach for non-parametric time-series classification, in: Proceedings of PKDD 2005, LNAI, vol. 3721, 2005, pp. 478–485.
[9] K. Yang, H. Yoon, C. Shahabi, CLeVer: a feature subset selection technique for multivariate time series, in: Proceedings of PAKDD 2005, LNAI, vol. 3518, 2005, pp. 516–522.
[10] C. Ding, J. Ye, Two-dimensional singular value decomposition (2DSVD) for 2D maps and images, in: Proceedings of SIAM International Conference on Data Mining (SDM'05), 2005, pp. 32–43.
[11] S. Hettich, S.D. Bay, The UCI KDD archive, University of California, Department of Information and Computer Science, Irvine, CA, 1999. Available from: <http://kdd.ics.uci.edu/>.
[12] B. Blankertz, G. Curio, K.R. Müller, Classifying single trial EEG: towards brain computer interfacing, in: T.G. Diettrich, S. Becker, Z. Ghahramani (Eds.), Advances in Neural Inf. Proc. Systems 14 (NIPS 01), 2002.
[13] Available from: <http://ida.first.fhg.de/projects/bci/competition_ii/>.
[14] J. Yang, D. Zhang, A.F. Frangi, J. Yang, Two-dimensional PCA: a new approach to appearance-based face representation and recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (1) (2004) 131–137.
[15] L. Wang, X. Wang, X. Zhang, J. Feng, The equivalence of two-dimensional PCA to line-based PCA, Pattern Recognition Letters 26 (2005) 57–60.
[16] L. Wei, E. Keogh, Semi-supervised time series classification, in: Proceedings of SIGKDD, 2006.
[17] E. Keogh, S. Kasetty, On the need for time series data mining benchmarks: a survey and empirical demonstration, in: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, pp. 102–111.
[18] R.T. Olszewski, Generalized Feature Extraction for Structural Pattern Recognition in Time-Series Data, Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, 2001.
[19] Available from: <http://www.cs.cmu.edu/~bobski/>.
[20] H. Zhang, T.B. Ho, M. Lin, X. Liang, Feature extraction for time series classification using discriminating wavelet coefficients, in: Proceedings of ISNN 2006, LNCS, vol. 3971, 2006, pp. 1394–1399.