JCSE Journal of Computing Science and Engineering

Regular Paper

Journal of Computing Science and Engineering,
Vol. 10, No. 2, June 2016, pp. 51-57

Influence of Data Preprocessing

Changming Zhu*
College of Information Engineering, Shanghai Maritime University, Shanghai, China and Department of Computer Science & Engineering, East China University of Science & Technology, Shanghai, China
cmzhu@shmtu.edu.cn

Daqi Gao
Department of Computer Science & Engineering, East China University of Science & Technology, Shanghai, China
gaodaqi@ecust.edu.cn

Abstract
In this paper, we research the influence of data preprocessing. We conclude that using different preprocessing methods leads to different classification performances. Moreover, not all data preprocessing methods are necessary, and a criterion is given to make sure which data preprocessing is necessary and which one is effective. Experiments on some real-world data sets validate that different data preprocessing methods result in different effects. Furthermore, experiments about some algorithms with different preprocessing methods also confirm that preprocessing has a great influence on the performance of a classifier.

Category: Smart and intelligent computing

Keywords: Data preprocessing; Preprocessing criterion; Fisher; Pseudo Inverse

I. INTRODUCTION

The number of patterns, dimensions, and classes are three basic characteristics for data sets. These characterizations have a great influence on the performance of classifiers, including matrix principal component analysis (MatPCA) [1], matrix Fisher linear discriminant analysis (MatFLDA) [1], support vector machine (SVM) [2], K-means algorithm [3], agglomerative hierarchical clustering (AHC) [4], improved multi-kernel classification machine with Nyström approximation technique (INMKMHKS) [5], nearest neighbor classifier (NNC) [6], K-nearest neighbor classifier (KNN) [7], Fisher [7], Pseudo Inverse [7], and some currently very popular spectral clustering approaches [8, 9]. However, none of them treats data preprocessing as a way to process recognition problems, especially when the distribution of data is very scattered. In this paper, we research the various data preprocessing methods and show that with the usage of an effective preprocessing method, a better classification performance and a structural invariance for a data set should be kept. Six main data preprocessing methods are widely used: normalized method, local rate change method, global rate change method, exponent change method, principal component analysis (PCA) method [10-12], and field-delete method. Not all data preprocessing methods are effective for any one data set; therefore, we propose a necessary method to determine the effectiveness of a data preprocessing method. In our paper, MatPCA, MatFLDA, SVM, NNC, K-means, AHC, INMKMHKS, NNC, KNN,

Open Access **http://dx.doi.org/10.5626/JCSE.2016.10.2.51** **http://jcse.kiise.org**

Received 19 March 2015; **Revised** 19 May 2016; **Accepted** 20 May 2016

*Corresponding Author

pISSN: 1976-4677 eISSN: 2093-8020

Fisher, and Pseudo Inverse are utilized to decide whether the structure is changed or not.

Other sections of the paper discuss the following criteria: In Section II, we give the description about some data preprocessing methods and the method for deciding whether a data preprocessing method is effective and meaningful; in Section III, the experimental results show the differences between these data preprocessing methods. Finally, both conclusion and future work are discussed in Section IV.

II. DATA PREPROCESSING

In order to process complicated data sets, six main data preprocessing methods can be employed: normalized method, local rate change method, global rate change method, exponent change method, PCA, and field-delete method.

1) Normalized method: If the covariance or the mean of data values in a dimension is very large, we should normalize data values of this dimension so as to make the center be zero.

2) Local rate change method: If some data values are so large or small in some dimensions, but in other dimensions the data values are similar, then a local rate change method may be a good choice.

3) Global rate change method: This method has a function similar to the local method, difference being that the local one changes each dimension by different maximum and minimum data values in each dimension while global one changes each dimension by the maximum and minimum data values from all data values.

4) Exponent change method: If the normalized method is unable to condense the data, then the exponent change method would be a good choice to reduce all data values. With the exponent change method, the range of data value is [0,1].

5) PCA: PCA is closely related to factor analysis; it uses a feature matrix to project the patterns into a new space. Namely, PCA can project patterns from a high-dimensional space into a low-dimensional space, where the patterns can be made to represent the original ones. The steps of PCA are as follows: (i) Compute the scatter matrix of patterns, (ii) compute the feature values (one feature value is a principal component) and feature vectors, (iii) sort feature values from large to small, (iv) choose more than 85% (in our method, it is 99% or 99.99%) of the principal components and combine the corresponding feature vectors as a projection matrix, (v) use the projection matrix to map the patterns in the original space into a new space, where the space dimension is based on the size of the corresponding feature vectors in this projection matrix.

6) Field-delete method: If more than 99% (including 100%) data values in some dimensions are zero, this dimension should be deleted since data values in this dimension do not have enough information for the classifier design. This method can reduce the dimension of the original patterns and simplify the classifier design.

Eqs. (1)–(4) give the basic computation methods of normalized method, local rate change method, global rate change method, and exponent change method, respectively.

$$x_{ij} = \frac{x_{ij} - x_{imean}}{\mathrm{var}_i} \tag{1}$$

$$x_{ij} = \frac{x_{ij} - x_{i\min}}{x_{i\max} - x_{i\min}} \tag{2}$$

$$x_{ij} = \frac{x_{ij} - x_{\min}}{x_{\max} - x_{\min}} \tag{3}$$

$$x_{ij} = \mathrm{e}^{(-x_{ij})} \tag{4}$$

where x_{ij} is j-th data value of i-th dimension, x_{imean} and var_i are the mean and covariance of data values in i-th dimension, $x_{i\min}$ and $x_{i\max}$ are the minimum and maximum data values of i-th dimension whereas $x_{\min}$ and $x_{\max}$ are the minimum and maximum data values of the whole data set.

For some recognition problems, we combine these methods for data preprocessing. Although different data preprocessing methods give different performances, not all of them give reasonable results. Some will change data structures and some won't. So here we adopt MatPCA, MatFLDA, SVM, NNC, K-means, AHC, INMKMHKS, NNC, KNN, Fisher, and Pseudo Inverse to judge whether a preprocessing method is reasonable or not. We define that if the number of testing patterns which are misclassified after data preprocessing has more than 1% distinction than the result without preprocessing, then this data preprocessing method is not effective; if the data structure is not very complicated, but we preprocess these patterns, then this preprocessing method is not meaningful. Here, a complicated data set means that a data set has a large covariance and mean, or has some very large or very small data values. Table 1 shows the used data preprocessing methods in our work.

III. EXPERIMENTS

In this section, we describe two experiments. First, we adopt MatPCA, MatFLDA, SVM, NNC, K-means, AHC, INMKMHKS, NNC, KNN, Fisher, and Pseudo Inverse to judge whether a data preprocessing method is reasonable or not, and whether the settings of their parameters can be referred to the respective reference. On the basis of the previous experimental results, we show, for these 21 data preprocessing methods (including the no preprocessing one), which techniques are better than others for different data sets. Second, we give the classification accuracies about these classifiers on different data sets when we

Table 1. Used data preprocessing methods

(1) No preprocessing
(2) Normalized
(3) Localized rate change
(4) Global rate change
(5) Localized rate change + exponent change
(6) Global rate change + exponent change
(7) Normalized + exponent change
(8) Field-delete (all zeros)
(9) PCA (99%)
(10) Field-delete (all zeros) + PCA (99%)
(11) Field-delete (all zeros) + PCA (99%) + normalized
(12) Field-delete (all zeros) + normalized
(13) Field-delete (99% zeros)
(14) Field-delete (99% zeros) + PCA (99%)
(15) Field-delete (99% zeros) + PCA (99%) + normalized
(16) Field-delete (99% zeros) + normalized
(17) PCA (99.99%)
(18) Field-delete (all zeros) + PCA (99.99%)
(19) Field-delete (all zeros) + PCA (99.99%) + normalized
(20) Field-delete (99% zeros) + PCA (99.99%)
(21) Field-delete (99% zeros) + PCA (99.99%) + normalized

PCA: principal component analysis.

adopt different data preprocessing methods. The main numerical characteristics of used UCI data sets [13] are summarized in Table 2.

A. Experimental Results of Data Preprocessing Methods for Different Data Sets

First, we discuss whether these data preprocessing methods are reasonable or not for different data sets. These experiments revealed that, no matter which classifier was used, the reasonability of a data preprocessing method for different data sets is similar. Thus, here we only discuss the reasonability of data preprocessing methods for these different data sets when NNC was used. Related experimental results are given in Fig. 1. Table 3 shows all average performances of the used classifiers with different data preprocessing methods used, so

Table 3. Average performances of classifiers with different data preprocessing methods used

Data processing (*M*)	**Classifier (*N*)**										
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
1	U	U	U	U	U	U	U	U	U	U	U
2	D	D	D	D	D	D	D	D	D	D	D
3	D	D	D	D	D	D	D	D	D	D	D
4	D	D	D	D	D	D	D	D	D	D	D
5	D	D	D	D	D	D	D	D	D	D	D
6	D	D	D	D	D	D	D	D	D	D	D
7	D	D	D	D	D	D	D	D	D	D	D
8	U	U	U	U	U	U	U	U	U	U	U
9	U	U	U	U	U	U	U	U	U	U	U
10	U	U	U	U	U	U	U	U	U	U	U
11	D	D	D	D	D	D	D	D	D	D	D
12	D	D	D	D	D	D	D	D	D	D	D
13	U	U	U	U	U	U	U	U	U	U	U
14	U	U	U	U	U	U	U	U	U	U	U
15	D	D	D	D	D	D	D	D	D	D	D
16	D	D	D	D	D	D	D	D	D	D	D
17	U	U	U	U	U	U	U	U	U	U	U
18	U	U	U	U	U	U	U	U	U	U	U
19	D	D	D	D	D	D	D	D	D	D	D
20	U	U	U	U	U	U	U	U	U	U	U
21	D	D	D	D	D	D	D	D	D	D	D

D: below the average performance level ($M<N$), U: over the average performance level ($M>N$).

Table 2. Basic characteristics of UCI data sets

Data sets	**No. of training**	**No. of testing**	**No. of classes**	**No. of dimensions**
2DVOW	338	333	10	2
Letter	16000	4000	26	16
Pen digit	7494	3498	10	16
Two spiral	194	194	2	2
Shuttle	43500	14500	7	9
Waveform	3332	1668	3	21
Pima Indians Diabetes (PID)	512	256	2	8

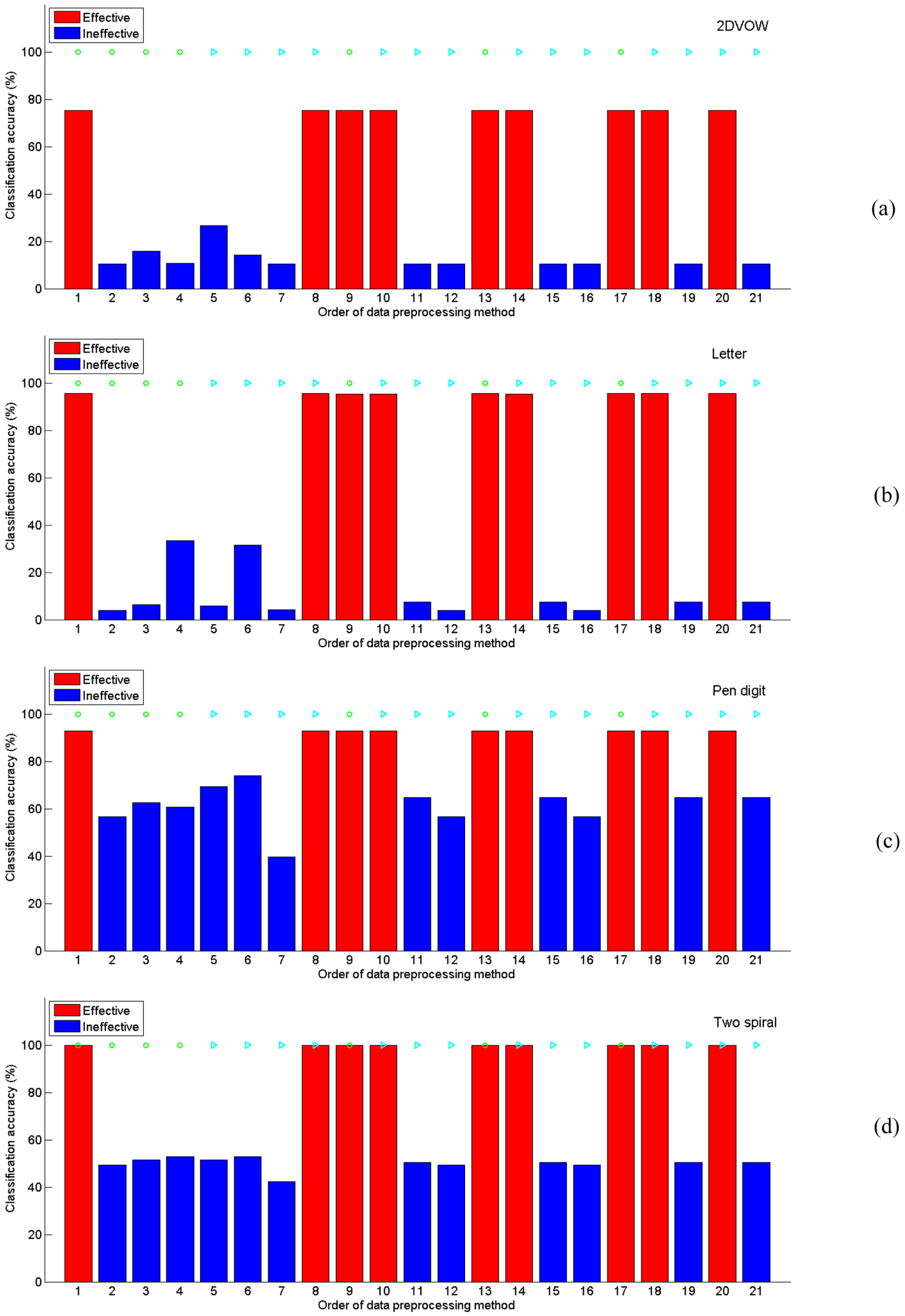

Fig. 1. Reasonability of data preprocessing methods for these different data sets when nearest neighbor classifier was used. (a) 2DVOW, (b) letter, (c) pen digit, (d) two spiral, (e) shuttle, (f) waveform, and (g) Pima Indians Diabetes (PID).

as to show, for these 21 data preprocessing methods (including the no preprocessing one), which techniques are better than others for different data sets.

In Fig. 1, the values of horizontal coordinate denote the 21 data preprocessing methods used, and one value corresponds to the order of one data preprocessing method where the orders are given in Table 1. The classification accuracy denotes the classification performance of NNC after data preprocessing. The red pillar denotes the corresponding data preprocessing method is effective, whereas the blue pillar represents ineffectiveness. The light green point represents that the corresponding data preprocessing method is meaningful whereas light blue triangle represents that the corresponding data preprocessing method

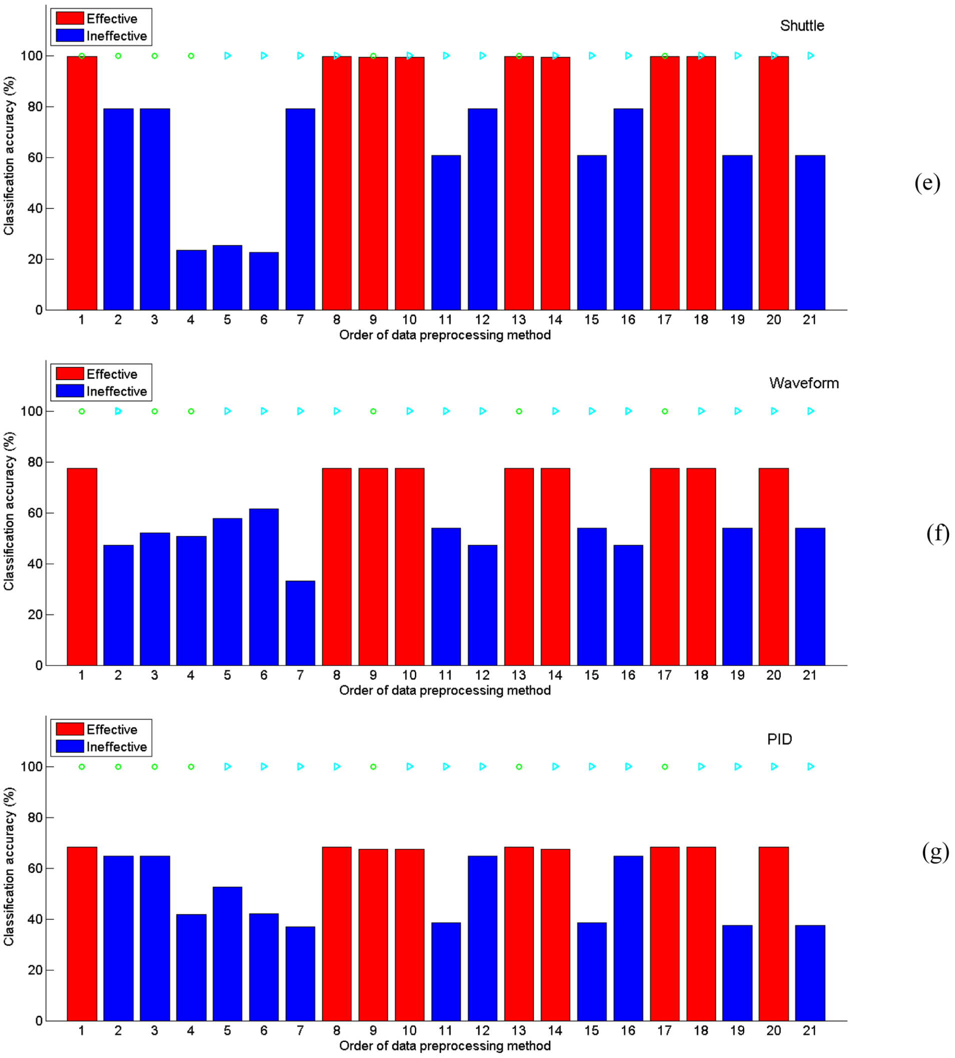

Fig. 1. Continued.

is meaningless. From Fig. 1, we see that the rate change methods are always meaningful but not effective for the structure of data that has been changed, while PCA does not change the relative relationship between patterns. Furthermore, we find that field-delete method is always effective but if we adopt exponent change method, it will be ineffective. Indeed, if too many data values are zeros in a dimension, we can delete this dimension since too many zero data do not have a great influence.

Additionally, in Table 3, we adopt 11 classifiers and 21 data preprocessing methods. For the row header, 1–11 adopt MatPCA, MatFLDA, SVM, NNC, K-means, AHC, INMKMHKS, NNC, KNN, Fisher, and Pseudo Inverse, respectively, while for the column header, 1–21 adopt the 21 data preprocessing methods. For each classifier and one data preprocessing method, we compute its average performance M on the 7 data sets used. Next, we compute the average performance N of this classifier with all Ms used. In Table 3, if M is smaller than N, i.e., the performance of this classifier with one data preprocessing method used is below the average level, we adopt D as a representation; if M is greater than N, we adopt U for representation. According to Table 3, we can draw a similar conclusion as the one derived from the experiments about NNC.

B. Classification Accuracies with Fisher and Pseudo Inverse Algorithm Used

Since the performances of the used classifiers are similar, here we give the results of the Fisher and Pseudo Inverse algorithms on different data sets after adopting data preprocessing methods shown in Fig. 2. This figure consists of two sub-figures. Fig. 2(a) shows the results when Fisher is adopted, whereas Fig. 2(b) shows the results when Pseudo Inverse is adopted. In each sub-figure, the values of horizontal coordinate denote 21 data preprocessing methods used; one value corresponds to the

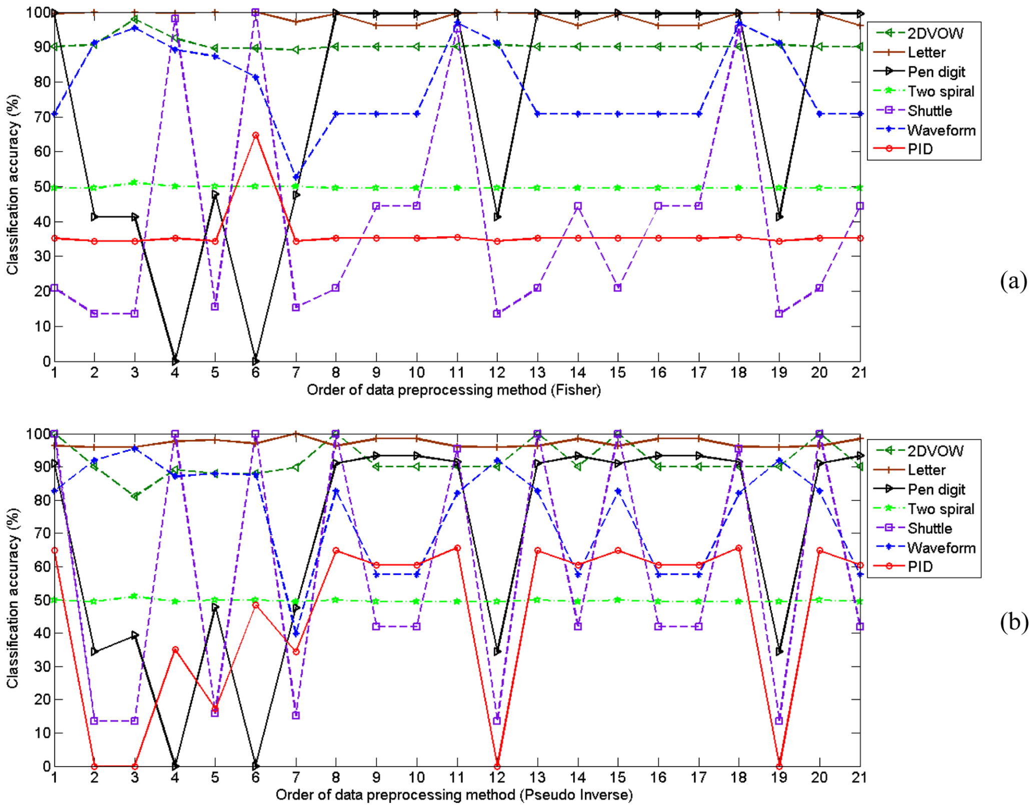

Fig. 2. Classification accuracies about (a) Fisher and (b) Pseudo Inverse algorithms on data sets using different data preprocessing methods.

order of one data preprocessing method. The orders of these data preprocessing methods are given in Table 1. In Fisher, we use $q = w^T(m_1\text{-}m_2)/2$ as the bias and $f(x) = w^Tx + q$ as a discriminate function where q is the bias, w is the destination vector, m_1 and m_2 are means of two class patterns. x is a test pattern. w is decided by $w = inv(S_{SW})*(m_1\text{-}m_2)^T$, where inv means the pseudo inverse computation, S_{SW} is the scatter matrix of two class patterns. In Pseudo Inverse, the bias and vector are shown in $C = inv(X)*Y$, here $C = [w_0, w]^T$ where w_0 is a bias and w is the destination vector. $f(x) = w^Tx + w_0$ is a discriminate function. X and Y represent the sets of patterns from two classes and their labels are selected in {-1,1}. For these two algorithms, one against one strategy is used.

From Fig. 2, we find that on most cases, when we use data preprocessing methods (1), (8), (9), (10), (13), (14), (17), (18), and (20), the average experimental results were always better. This conclusion is similar to what we found in Section III-A.

IV. CONCLUSION AND FUTURE WORK

In our work, we discuss the influence of data preprocessing methods for different data sets. We also give a definition to judge whether a data preprocessing method is reasonable or not. Our study showed that for different data sets, different data preprocessing methods bring different results, and that they affect the performances of classifiers.

In the future, we will make full use of the influence of data preprocessing and apply them to other algorithms. Doing this, we want to improve the classification performances about other classical algorithms.

ACKNOWLEDGMENTS

This work was partially supported by Shanghai Natural Science Foundation under Grant No. 16ZR1414500; Natural Science Foundations of China under Grant No. 61272198 and 21176077; Innovation Program of Shanghai Municipal Education Commission under Grant No. 14ZZ054; the Fundamental Research Funds for the Central Universities, Shanghai Key Laboratory of Intelligent Information preprocessing of China under Grant No. IIPL-2012-003; Soochow University Jiangsu Provincial Key Laboratory for Computer Information preprocessing Technology, and Nature Science Foundation of Shanghai Province of China under Grant No. 11ZR1409600.

REFERENCES

1. S. Chen, Y. Zhu, D. Zhang, and J. Y. Yang, "Feature extraction approaches based on matrix pattern: MatPCA and MatFLDA," *Pattern Recognition Letters*, vol. 26, no. 8, pp. 1157-1167, 2005.
2. N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge: Cambridge University Press, 2000.
3. J. A. Hartigan and M. A. Wong, "Algorithm AS 136: a k-means clustering algorithm," *Journal of the Royal Statistical Society Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100-108, 1979.
4. A. J. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Englewood Cliffs, NJ: Prentice-Hall Inc., 1988.
5. C. Zhu, "Improved multi-kernel classification machine with Nyström approximation technique and Universum data," *Neurocomputing*, vol. 175A, pp. 610-634, 2016.
6. V. N. Vapnik, *Statistical Learning Theory*, New York: Wiley, 1998.
7. E. Fix and J. L. Hodges, "Discriminatory analysis: nonparametric discrimination: consistency properties," *International Statistical Review*, vol. 57, no. 3, pp. 238-247, 1989.
8. A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: analysis and an algorithm," *Advances in Neural Information Processing Systems*, vol. 2, pp. 849-856, 2002.
9. S. X. Yu and J. Shi, "Multiclass spectral clustering," in *Proceedings of 9th IEEE International Conference on Computer Vision*, Nice, France, 2003, pp. 313-319.
10. K. Person, "On lines and planes of closest fit to system of points in space," *Philiosophical Magazine Series 6*, vol. 2, no. 11, pp. 559-572, 1901.
11. I. T. Jolliffe, *Principal Component Analysis*, New York: Springer, 2002.
12. C. Saunders, J. Shawe-Taylor, and A. Vinokourov, "String kernels, fisher kernels and finite state automata," *Advances in Neural Information Processing Systems*, vol. 15, pp. 649-656, 2003.
13. D. J. Newman, S. Hettich, C. L. Blake, C. J. Merz, and D. W. Aha, "UCI repository of machine learning databases," 1998; http://archive.ics.uci.edu/ml/datasets.htm.

Changming Zhu

Changming Zhu received the B.Sc. degree and Ph.D. degree in Department of Computer Science and Engineering, East China University of Science and Technology, China, in 2010 and 2015, respectively. Currently he is a teacher in Shanghai Maritime University. His research interests are in neural computing and pattern recognition.

Daqi Gao

Daqi Gao received the Ph.D. degree from Zhejiang University, China, in 1996. Currently, he is a Professor in East China University of Science and Technology. He is a member of the International Neural Network Society. He has published over 50 scientific papers. His research interests are in pattern recognition, neural networks, and machine olfactory.