Hindawi Publishing Corporation
Advances in Artificial Intelligence
Volume 2012, Article ID 484595, 7 pages
doi:10.1155/2012/484595

Research Article

RPCA: A Novel Preprocessing Method for PCA

Samaneh Yazdani,[1] Jamshid Shanbehzadeh,[2] and Mohammad Taghi Manzuri Shalmani[3]

[1] *Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran*
[2] *Department of Computer Engineering, Faculty of Engineering, Kharazmi University, Tehran, Iran*
[3] *Electronic Research Center, Sharif University of Technology, Tehran, Iran*

Correspondence should be addressed to Samaneh Yazdani, samaneh.yazdani@gmail.com

Received 15 May 2012; Revised 27 September 2012; Accepted 5 November 2012

Academic Editor: Wolfgang Faber

We propose a preprocessing method to improve the performance of Principal Component Analysis (PCA) for classification problems composed of two steps; in the first step, the weight of each feature is calculated by using a feature weighting method. Then the features with weights larger than a predefined threshold are selected. The selected relevant features are then subject to the second step. In the second step, variances of features are changed until the variances of the features are corresponded to their importance. By taking the advantage of step 2 to reveal the class structure, we expect that the performance of PCA increases in classification problems. Results confirm the effectiveness of our proposed methods.

1. Introduction

In many real world applications, we faced databases with a large set of features. Unfortunately, in the high-dimensional spaces, data become extremely sparse and far apart from each other. Experiments show that in this situation once the number of features linearly increases, the required number of examples for learning exponentially increases. This phenomenon is commonly known as the curse of dimensionality. Dimensionality reduction is an effective solution to the problem of curse of dimensionality [1, 2]. Dimensionality reduction is to extract or select a subset of features to describe the target concept. The selection and extraction are based on finding a relevant subset of original features and generating a new feature space through transformation, respectively [1, 3]. The proper design of selection or extraction process improves the complexity and the performance of learning algorithms [4].

Feature selection concerns representing the data by selecting a small subset of its features in its original format [5]. The role of feature selection is critical, especially in applications involving many irrelevant features. Given a criterion function, feature selection is reduced to a search problem [4, 6]. Exhaustive search, when the number of the features is too large, is infeasible and heuristic search can be employed. These algorithms, such as sequential forward and/or backward selection [7, 8], have shown successful results in practical applications. However, none of them can provide any guarantee of optimality. This problem can be alleviated by using feature weighting, which assigns a real-value number to each feature to indicate its relevancy to the learning problem [6]. Among the existing feature weighting algorithms, ReliefF [5] is considered as one of the most successful ones due to its simplicity and effectiveness [9]. A major shortcoming of the feature weighting is its inability to capture the interaction of correlated features [4, 10]. This drawback can be solved by some feature extraction techniques.

The basis of feature extraction is a mathematical transformation that changes data from a higher dimensional space into a lower dimensional one. Feature extraction algorithms are generally effective [11]. However, their effectiveness will be degraded when they are used for processing large-scale datasets [12]. In addition, the features extracted from the mathematical transformation usually concern with all original features. So the extracted features may contain

ReliefF Algorithm

(1) Initialization: given $D = \{(x_j, y_j)\}_{j=1}^{N}$, y is the label of classes between $1 \dots c$.
c number of class, set $w_i = 0$, $1 \le i \le t$, number of iteration T;
(2) for $l = 1$ to T
(3) Randomly select a pattern x_r from D with class y_r;
(4) Find k nearest hits H_j from class y_r
(5) For each class $y \neq y_r$
(6) from class y find k nearest misses $M_j(y)$
(7) For $i = 1$ to t
(8) compute: $w_i = w_i - \sum_{j=1}^{k} \frac{|x_{ri} - H_{ji}|}{T \cdot k} + \sum_{y \neq y_r} \left(\frac{p(y)}{1 - p(y_r)} \sum_{j=1}^{k} \frac{|x_{ri} - M_{ji}(y)|}{T \cdot k} \right)$
(9) end
(10) end

Pseudocode 1: Pseudocode of ReliefF [2].

information originated from the irrelevant information in the original space [3, 13].

Principal Component Analysis (PCA) is an effective feature extraction approach and has successfully been applied in recognition applications such as face, handprint, and human-made object recognition [14–16] and industrial robotics [17]. The traditional PCA is an orthogonal linear transformation and operates directly on a whole pattern represented as a vector and acquires a set of projections to extract global feature from a given training pattern [18]. PCA reduces the dimension such that the representation is as faithful as possible to the original data [2]. PCA employs all features in the original space, regardless their relevancy, to produce new features. This may result in features containing information originated from irrelevant features in the original space. A side effect is misclassification results. Some works have been done to improve the performance of PCA via the feature weighting. In [19, 20], feature weighting has been used for eliminating irrelevant features or using the weight of features in its calculation. In [19], rank is used instead of the original data for copying the outliers and noises. Honda et al. used weights of features in PCA-guided formulation, while in our proposed method we utilize weights of features to properly change the dataset.

The main objective of this paper is to improve the accuracy of classification using features extracted by PCA. PCA is the best-known unsupervised linear feature extraction algorithm; but it is used for classification tasks too. Since PCA do not pay any particular attention to the underlying class structure, it is not always an optimal dimensionality-reduction procedure for classification purposes, and the projection axes chosen by PCA might not provide the good discrimination power. However, the study in [21] illustrates that PCA might outperform LDA which is one of the best supervised dimensionality reduction method, when the number of samples per class is small or when the training data nonuniformly samples the underlying distribution. In the present work, we propose a novel preprocessing method composed of two steps. In the first step, the qualities of features are computed via a feature weighting algorithm. The selected relevant features, features with weights larger than a predefined threshold, are then subject to the second step. In the second step, the variances of features are modified until the most relevant ones become the most important ones for PCA. Finally, PCA is performed on them to generate uncorrelated features.

The rest of this paper is organized as follows. Section 2 reviews ReliefF, PCA, and its associated problems in brief. Section 3 describes the proposed algorithm. Section 4 presents our experiments on both synthetic and real data and the final section is Conclusion.

2. Review of the ReliefF and PCA Methods

This section reviews ReliefF and PCA briefly and presents the drawbacks of PCA.

2.1. ReliefF. Relief [5] is one of the most successful algorithms to assess the quality of features. The main idea of Relief is to iteratively estimate the weights of features according to how well values distinguish among instances that are near each other. The original Relief limits into two classes problems and deals with complete data [22]. In particular, it has no mechanism to eliminate redundant features [23]. This paper utilizes an extension of Relief called ReliefF [22] that solves the two first problems of Relief. In contrast to Relief, which uses the 1-nearest-neighbor algorithm, ReliefF uses an approach based on K-nearest-neighbor algorithms. Pseudocode 1 presents the pseudocode of this algorithm. It is assumed that $D = \{(x_j, y_j)\}_{j=1}^{N}$ denotes a training dataset with N samples in which each sample consists of t features $x = (x_1, \dots, x_t)$ and the known class label y_j. In each iteration, ReliefF randomly selects a sample (pattern) x and then searches k of its nearest neighbors from the same class, termed nearest hits H_j, and also the nearest neighbors from each of different classes, called nearest misses $M_j(y)$. To compute the weight of each feature, ReliefF uses the contribution of all the hits and misses.

Table 1: Centroids and standard deviations of classes in different variables.

Class	Class centroids	Standard deviations	No. of points
1	(0.547, 0728, 0.424, 0.492, 0561)	(0.054, 0.044, 0.071, 0.288, 0.302)	100
2	(0.299, 0.585, 0.318, 0.555, 0.455)	(0.061, 0.044, 0.069, 0.269, 0.274)	100
3	(0.422, 0.452, 0.636, 0.520, 0.536)	(0.055, 0.050, 0.075, 0.263, 0.274)	100

In ReliefF algorithm, T is a parameter defined by users and determines the number of process repeats to estimate the weight of each feature. x_{ri} is the ith feature of sample x_r and $p(y)$ is the prior probability of class y.

2.2. Principle Component Analysis. PCA is a very effective approach of extracting features. It is successfully applied to various applications of pattern recognition such as face classification [18]. As mentioned above, N and t are the number of samples and their dimension of dataset D, respectively. PCA finds a subspace whose basis vectors correspond to the maximum-variance direction of the original space. As mentioned before, PCA is a linear transform. Let W represents the linear transformation that maps the original t-dimensional space into an f-dimensional feature space where normally $f \ll t$. Equation (1) shows the new feature vectors, $z_j \in R^f$

$$z_j = W^T x_j, \quad j = 1, 2, \ldots, N. \tag{1}$$

Columns of W are the eigenvectors e_i obtained by solving (2):

$$\lambda_j e_j = Q e_j \quad \text{where } Q = XX^T,\ X = \{x_1, \ldots, x_N\}. \tag{2}$$

Here Q is the covariance matrix and λ_j the eigenvalue associated with the eigenvector e_j. The eigenvectors are sorted from high to low according to their corresponding eigenvalues. The eigenvector associated with largest eigenvalue is the most important vector that reflects the greatest variance [21].

PCA employs the entire features and it acquires a set of projection vectors to extract global feature from given training samples. The performance of PCA is reduced when there are more irrelevant features than the relevant ones. On the other hand, PCA has no preknowledge about the class in a given data. So, it is not efficient to determine the classes in the subspace of a given dataset.

We present an example to confirm the mentioned points. This example uses a dataset with five variables and 300 records. The number of classes is three and each class has 100 points. The last two variables represent uniform distributed noise points and irrelevant features. Table 1 shows the centroids and the standard deviations of the three classes [24].

The centroids of two noise variables (x_3 and x_4), against other three variables, are very close and their standard deviations are larger than those of the other three variables. Figure 1, illustrates the 300 points in different two-dimensional subspaces. We can find no class structure in subspaces with two noisy features. Now, PCA is applied on the database presented in Table 1. Figure 2 shows the results obtained by using two significant eigenvectors extracted by PCA.

Figure 2 shows that the obtained result is not suitable for classification, because there is no mechanism in PCA algorithm to determine irrelevant features. As mentioned before, PCA finds projections of the data with maximum variance. Observably, in this example, there are two irrelevant features with the largest variance. Now, PCA is just performed on three relevant variables x_1, x_2, x_3. Figure 3 illustrates the new data by applying the PCA. Notice that the class structures can be found in Figure 3. Because of removing irrelevant features, it is suitable for classification. The next section presents the proposed algorithm to solve this problem.

3. RPCA Feature Extraction

As shown in Figure 2, the directions founded by PCA are not proper for classification if the variances of features are not corresponding with their importance. For example, if the variances of irrelevant features are large, then the extracted features via PCA are not suitable for classification. Therefore, it is expected that if the importance of features are proper with their variances then the extracted features using PCA are more likely suitable for classification. In this paper, a new preprocessing method is proposed which involves two connected steps: relevance analysis and variance adjustment as shown in Figure 4.

In the step of the relevant analysis, weights of features are calculated through one feature weighting approach (like Relief or its extension for multiclass dataset called ReliefF). Assume that $W = [w_1, w_2, \ldots, w_t]$ be the weight vector, estimated by using ReliefF, for the t variables in the original space. Since the weights indicate the level of relevancy, the feature with the largest weight has the largest relevancy. The relevancy level is close to zero or negative when the feature is irrelevant [5]. In this work, features with the weights larger than the threshold defined by user γ are the subject to the next step. Therefore, W vector is changed as follows:

$$w_i = \begin{cases} w_i & w_i > \gamma, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

After removing the irrelevant features, we do not need to collect all the features. In the variance adjustment step, the variances of features have been changed so that the most important feature becomes the most important feature for PCA. A key idea for this step is motivated from this characteristic of PCA: a feature with maximum variance has the most important for PCA. The new variance of ith feature is calculated as follows:

$$\delta_{\text{new}i} = m - (w_{k(m)} - w_i)(m - k(i)), \tag{4}$$

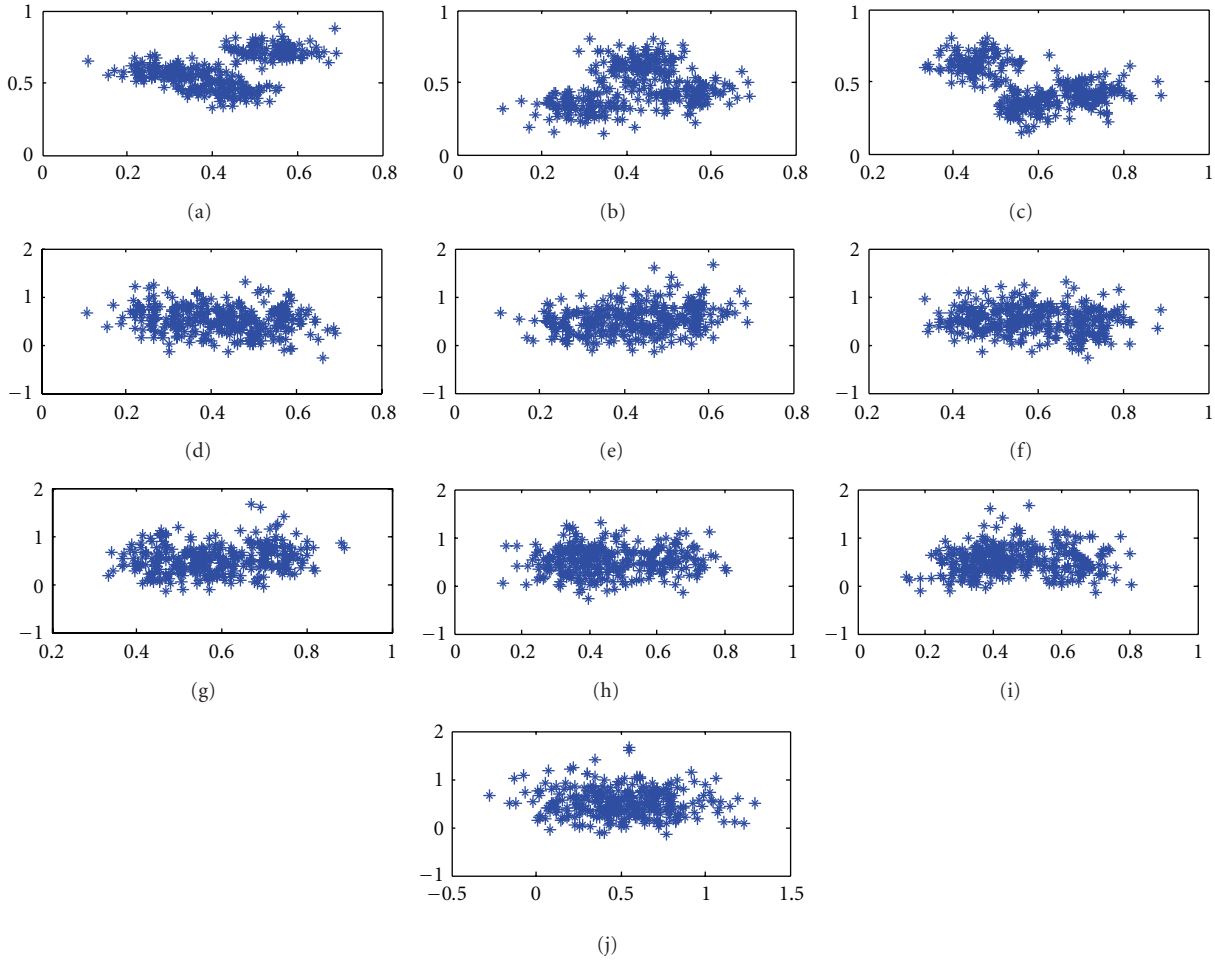

Figure 1: Synthetic dataset with three normally distributed classes in the three-dimensional subspace of x_0, x_1, x_2 and two noise variables x_3, x_4. (a) The subspace of x_0, x_1. (b) The subspace of x_0, x_2. (c) The subspace of x_1, x_2. (d) The subspace of x_0, x_3. (e) The subspace of x_0, x_4. (f) The subspace of x_1, x_3. (g) The subspace of x_1, x_4. (h) The subspace of x_2, x_3. (i) The subspace of x_2, x_4. (j) The subspace of x_3, x_4 [24].

where m is the number of features that their weight are more than threshold (number of relevant features). $w_{k(m)}$ is the weight of most important feature and $k(i)$ is the weight rank of i-th feature (1 is least importance and m is most importance). Since $w_{k(m)} - w_i \succ 0$, and $(m - k(i)) \geq 0$, $\delta_{\text{new}i}$ is always positive. It is important to mention that $w_{k(m)} \succ w_i$ because $w_{k(m)}$ is the largest weight. Then, to modify the variance of i-th feature to $\delta_{\text{new}i}$, the values of it should be multiplied by the number specified for it. So, it is calculated as follows:

$$\begin{aligned}
\delta_{\text{new}i} &= \frac{1}{N-1}\sum_{j=1}^{N}\left(nx_{ji} - n\vec{x}_i\right)^2,\\
\delta_{\text{new}i}(N-1) &= \sum_{j=1}^{N}\left(nx_{ji} - n\vec{x}_i\right)^2,\\
n &= \sqrt{\frac{\delta_{\text{new}i}(N-1)}{\sum_{j=1}^{N}\left(x_{ji} - \vec{x}_i\right)^2}}.
\end{aligned} \tag{5}$$

Equation (5) shows the way that can obtain n for each feature where $\sigma_{\text{new}i}$ is the new variance of ith feature and calculated using (4). N is the number of samples and $x_{ji}, \vec{x}_i$ are ith feature of jth sample and mean of ith feature, respectively. After this adjustment, PCA is employed on data. We call our proposed method RPCA that refers to applying ReliefF in the first step for weighting features.

Notice that each feature weighting method can be utilized in the first step. Since the output of the first step is used as a subject for the second step (variance adjustment), more effective feature weighting methods lead to better results. Hence, if we use a feature weighting more effective than ReliefF, the obtained result is better than we use ReliefF. Further, the type of feature weighting is very important. For example, if we replace ReliefF with another unsupervised feature weighting method like SUD [25], the proposed method can be utilized for the unsupervised dataset as a dimensionality reduction. The advantages of our preprocessing method are summarized as follows.

(i) The extracted features are formed only by using relevant features.

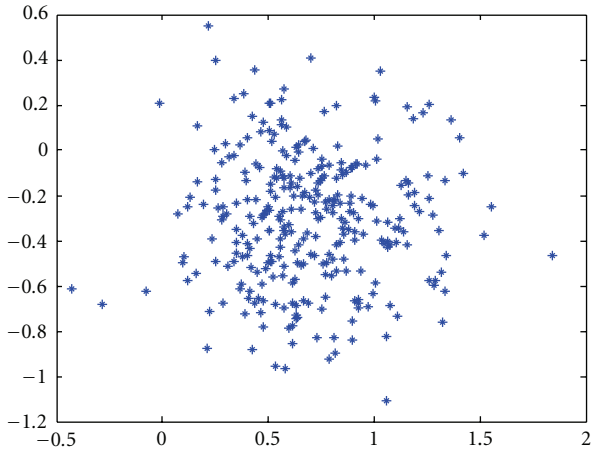

Figure 2: A plot of a new data point by applying the PCA using two significant eigenvectors.

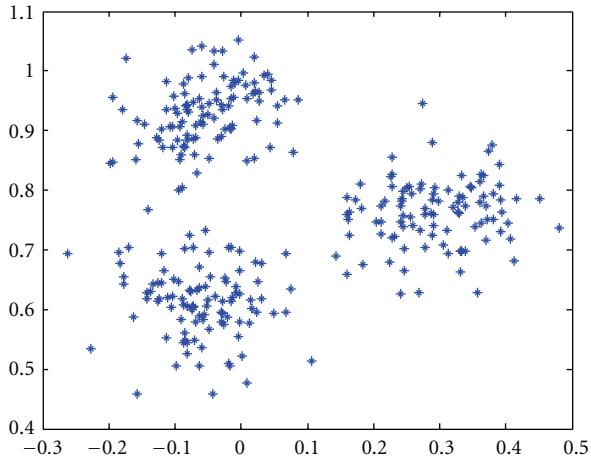

Figure 3: A new data point by applying the PCA using two significant eigenvectors after removing irrelevant features.

(ii) The preprocessing steps have low time complexity.

(iii) The preprocessing steps reveal the underlying class structure for PCA approximately.

4. Simulation Results

This section presents the experimental results to show the effectiveness of RPCA on four UCI datasets and synthetic data introduced in Section 2.2. Table 2 summarizes the data information of the four UCI datasets. We applied ReliefF, which employs M instead of just one nearest hit and miss, in our experiment. The value of M was set to 10 as suggested in [22].

In order to provide a platform where PCA and RPCA can be compared, KNN classification errors are used. The number of nearest neighbors is achieved by trial and error. To eliminate statistical variation, each algorithm is run 20 times for each dataset. In each run, a dataset is randomly

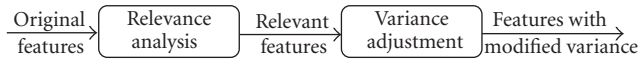

Figure 4: Proposed preprocessing steps.

Table 2: Summary of four UCI data sets.

Database	Training	Testing	Features
Twonorm	400	7000	20
Waveform	400	4600	21
Ringnorm	400	7000	20
Breast cancer	100	545	9

Table 3: The testing errors.

Database	PCA	RPCA
Synthetic data	0.5787	0.0083
Twonorm	0.2529	0.0349
Waveform	0.6653	0.2496
Ringnorm	0.5021	0.1797
Breast cancer	0.3581	0.0434

partitioned into training and testing. Also, 50 irrelevant features with Gaussian distributions are added to UCI datasets. The mean of Gaussian distribution is equal to zero and the standard deviation is set based on dataset.

Table 3 shows the testing errors. The number of extracted features is five expected in syntactic dataset which is two in this dataset. The number of training and testing instances for synthetic dataset are 100 and 200, respectively. The performance of KNN is degraded significantly in the presence of the large number of irrelevant features [6]. Figure 5 illustrates the average testing errors of PCA and RPCA as a function of the number of extracted features for 20 runs. This figure reveals that RPCA significantly outperforms PCA in terms of classification errors and effectiveness in reducing dimensionality. These results show that RPCA can significantly improve the performance of KNN. As discussed in Section 3, using a feature weighting better than ReliefF in the first step can lead to better results.

5. Conclusion

We propose a new preprocessing method comprised two steps to improve the performance of PCA in classification task. After weighting features and selecting relevant features in the first step, the variances of features are adjusted based on their importance in the second step until the most important feature has the most variance. Finally, PCA is applied to the modified data. Since, in the first step, ReliefF is used for feature weighting, we nominate our proposed preprocessing technique RPCA. Moreover, we can utilize another type of feature weighting method instead of ReliefF. For example, SUD [25] can be employed in unsupervised data. The simulation results show that the RPCA significantly improves the efficiency of PCA in classification purposes.

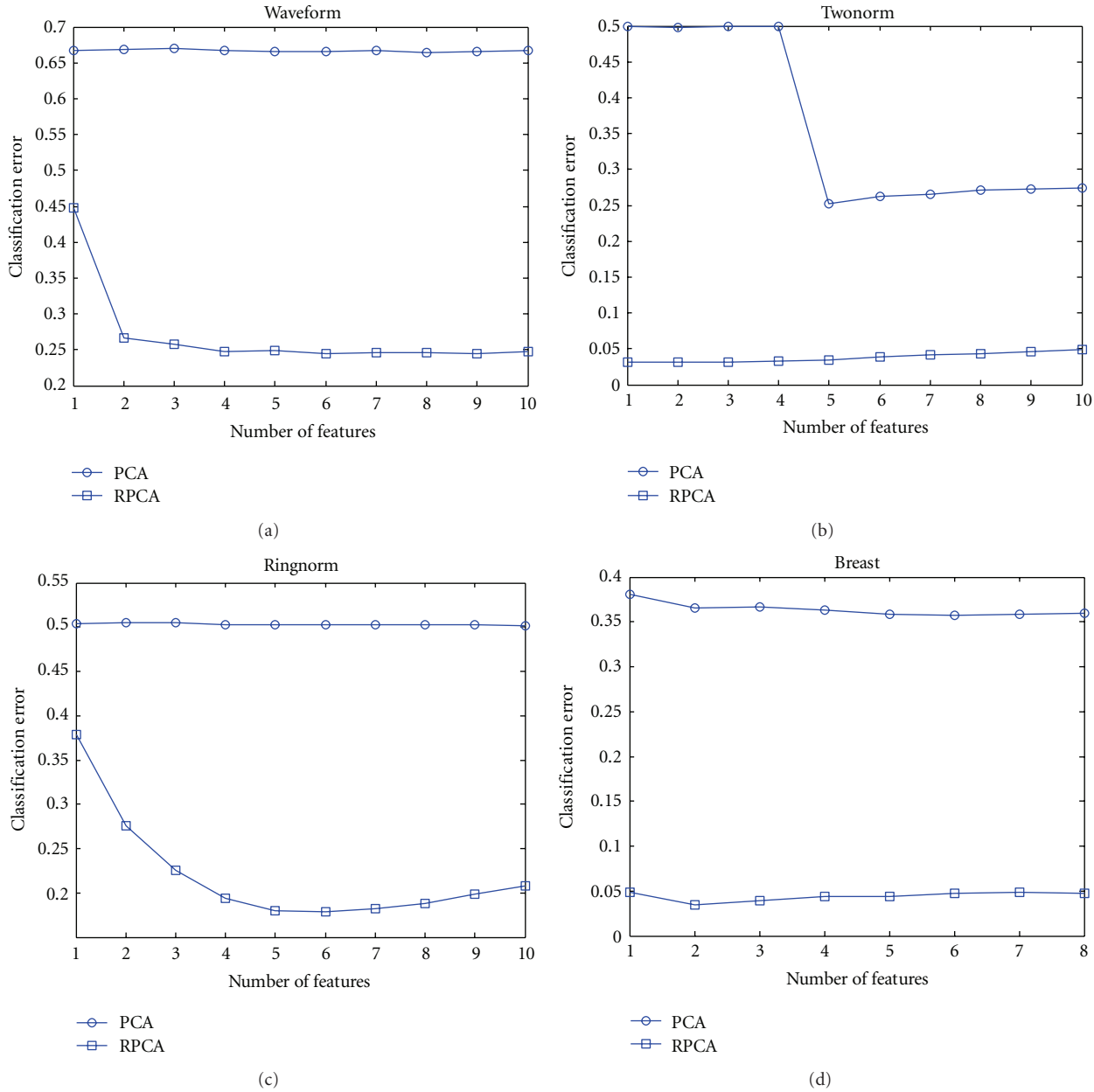

Figure 5: Classification errors of PCA and RPCA on the four UCI datasets.

Acknowledgment

This research is supported by Iran Telecommunication Research Center (ITRC).

References

[1] M. Dash and H. Liu, "Dimensionality reductionin," in *Encyclopedia of Computer Science and Engineering*, B. W. Wah, Ed., vol. 2, pp. 958–966, John Wiley & Sons, Hoboken, NJ, USA, 2009.

[2] H. Liu and H. Motoda, *Computational Methods of Feature Selection*, Taylor & Francis Group, 2008.

[3] M. Yang, F. Wang, and P. Yang, "A novel feature selection algorithm based on hypothesis-margin," *Journal of Computers*, vol. 3, no. 12, pp. 27–34, 2008.

[4] Y. Sun and D. Wu, "A RELIEF based feature extraction algorithm," in *Proceedings of the 8th SIAM International Conference on Data Mining*, pp. 188–195, April 2008.

[5] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Proceedings of the 9th International Conference on Machine Learning*, pp. 249–256, 1992.

[6] Y. Sun, "Iterative RELIEF for feature weighting: algorithms, theories, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1035–1051, 2007.

[7] S. C. Yusta, "Different metaheuristic strategies to solve the feature selection problem," *Pattern Recognition Letters*, vol. 30, no. 5, pp. 525–534, 2009.
[8] P. Pudil and J. Novovičová, "Novel methods for subset selection with respect to problem knowledge," *IEEE Intelligent Systems and Their Applications*, vol. 13, no. 2, pp. 66–74, 1998.
[9] T. G. Dietterich, "Machine-learning research: four current directions," *AI Magazine*, vol. 18, no. 4, pp. 97–136, 1997.
[10] D. Wettschereck, D. W. Aha, and T. Mohri, "A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms," *Artificial Intelligence Review*, vol. 11, no. 1–5, pp. 273–314, 1997.
[11] J. Yan, B. Zhang, N. Liu et al., "Effective and efficient dimensionality reduction for large-scale and streaming data preprocessing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 3, pp. 320–332, 2006.
[12] R. Jensen and Q. Shen, *Computational Intelligence and Feature Selection Rough and Fuzzy Approach*, Press Series on Computational Intelligence, John Wiley & Sons, 2008.
[13] I. T. Jolliffe, *Principal Component Analysis*, Wiley, 2nd edition, 2002.
[14] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
[15] H. Murase, F. Kimura, M. Yoshimura, and Y. Miyake, "An improvement of the autocorrelation matrix in pattern matching method and its applicationto handprinted 'HIRAGANA'," *IECE Transactions*, vol. 64, no. 3, pp. 276–283, 1981.
[16] H. Murase and S. K. Nayar, "Visual learning and recognition of 3-d objects from appearance," *International Journal of Computer Vision*, vol. 14, no. 1, pp. 5–24, 1995.
[17] S. K. Nayar, S. A. Nene, and H. Murase, "Subspace methods for robot vision," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 750–758, 1996.
[18] S. Chen and Y. Zhu, "Subpattern-based principle component analysis," *Pattern Recognition*, vol. 37, no. 5, pp. 1081–1083, 2004.
[19] J. F. Pinto Da Costa, H. Alonso, and L. Roque, "A weighted principal component analysis and its application to gene expression data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 8, no. 1, pp. 246–252, 2011.
[20] K. Honda, A. Notsu, and H. Ichihashi, "Variable weighting in PCA-guided κ-means and its connection with information summarization," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 15, no. 1, pp. 83–89, 2011.
[21] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228–233, 2001.
[22] I. Kononenko, "Estimating attributes: analysis and extensions of RELIEF," in *Proceedings of the European Conference on Machine Learning (ECML '54)*, pp. 71–182, 1994.
[23] R. Gilad-Bachrach, A. Navot, and N. Tishby, "Margin based feature selection—theory and algorithms," in *Proceeding of the 21st International Conference on Machine Learning (ICML '04)*, pp. 337–344, July 2004.
[24] J. Z. Huang, M. K. Ng, H. Rong, and Z. Li, "Automated variable weighting in k-means type clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 657–668, 2005.
[25] M. Dash, H. Liu, and J. Yao, "Dimensionality reduction of unsupervised data," in *Proceedings if the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI '97)*, pp. 532–539, November 1997.