Training Data Reduction and Classification Based on Greedy Kernel Principal Component Analysis and Fuzzy C-means Algorithm

Xiaofang Liu
School of Computer Science
Sichuan University of Science and Engineering
Zigong, China
lxf1969@163.com

Chun Yang
School of Economics and Management
Sichuan University of Science and Engineering
Zigong, China
yangchun1972@163.com

***Abstract*—Nonlinear feature extraction used standard Kernel Principal Component Analysis (KPCA) method has large memories and high computational complexity in large datasets. A Greedy Kernel Principal Component Analysis (GKPCA) method is applied to reduce training data and deal with the nonlinear feature extraction problem for training data of large data in classification. First, a subset, which approximates to the original training data, is selected from the full training data using the greedy technique of the GKPCA method. Then, the feature extraction model is trained by the subset instead of the full training data. Finally, FCM algorithm classifies feature extraction data of the GKPCA, KPCA and PCA methods, respectively. The simulation results indicate that the feature extraction performance of both the GKPCA, and KPCA methods outperform the PCA method. In addition of retaining the performance of the KPCA method, the GKPCA method reduces computational complexity due to the reduced training set in classification.**

Keywords-training data reduction; classification; nonlinear feature extraction; greedy kernel principal component analysis; fuzzy C-means algorithm; kernel matrix

I. INTRODUCTION

The Kernel Principal Component Analysis (KPCA) method in [1] is the nonlinear extension of the ordinary linear Principal Component Analysis (PCA) method. It shows a powerful nonlinear feature extraction technique by kernel methods in [2]. The kernel methods use kernel functions to perform the feature space straightening effectively. This technique allows to using established theory behind the linear algorithms to design their nonlinear counterparts. A disadvantage of the KPCA method in [3], however, is that the storage of training data in terms of the dot products is too expensive since the size of kernel matrix increases quadratically with the number of training data. The standard KPCA method could process limited number of training data. For large scale data set, it suffers from computational problem of diagonal and occupies large storage space of kernel matrix. The size of training data is therefore vital in any real application incorporating the KPCA method. So, a more efficient feature extraction method, Greedy Kernel Principal Component Analysis (GKPCA) method in [4] is applied to reduce training data and nonlinear feature extraction for classification.

II. GREEDY KERNEL PRINCIPAL COMPONENT ANALYSIS METHOD

The standard KPCA method is an extension of the PCA method by kernel methods, whereby the data in the input space $\aleph$ are nonlinearly mapped to a feature space F via some nonlinear map Φ before applying the PCA method. Give a set of training data $X = \{x_1, x_2, \cdots, x_N\} \subset R^q$, N is the number of samples, q is the dimension of the sample x_i. The data set X, $x_i \in \aleph \subset R^q$ $(i = 1, 2, \cdots, N)$ are mapped by a function $\Phi : \aleph \to F$ to a new high dimensional feature space F. Note feature space F could have an arbitrarily large, possibly infinite dimensionality. The PCA method is applied on the mapped data $X_\Phi = \{\Phi(x_1), \Phi(x_2), \cdots, \Phi(x_N)\}$. The computation of the principal components and the projection on these components can be expressed in terms of dot products thus the kernel functions $K : \aleph \times \aleph \to R$ can be employed.

A disadvantage of the KPCA method in [3], however, is that the training and evaluation costs are dependant on the size of the training data. During training, an $N \times N$ size of the kernel matrix K, which grows quadratically with the number N of samples in the training data, needs to be calculated before the PCA method can be applied in feature space. The size of the training data is therefore vital in any real system incorporating the KPCA method.

The GKPCA method in [4] is proposed by V. France to reduce training set in 2005. It is an efficient algorithm to compute the ordinary KPCA method. The approach aims to represent data in a low dimensional space with possibly minimal representation error which is similar to the PCA method. In contrast to the PCA method, the basis vectors of the low dimensional space used for data representation are properly selected vectors from the training data and not as their linear combinations.

Let $X = \{x_1, x_2, \cdots, x_N\} \subset R^q$ be the set of input training data. The vector set $X_S = \{s_1, s_2, \cdots, s_n\} \subset R^q$, whose size is much smaller than that of training data X, is a subset of training data X. Let $J = \{j_1, j_2, \cdots, j_n\}$ be the indices of subset X_S in [5], a subset of X, where $I = \{i_1, i_2, \cdots, i_N\}$ is the original indices of X. The reduced set method aims to

find a new kernel expansion and well approximates the original one. The approximate feature space representation of the original training samples can be expressed as follows:

$$\tilde{\Phi}(x_i) = \sum_{j \in J} \beta_{ij} \Phi(x_j), \forall i \in I. \quad (1)$$

The problem of finding the reduced kernel expansion can be stated as the optimization task. The objective of the GKPCA method is to minimize the mean square error in (2) while the size n of the subset X_S is kept small. The GKPCA method is implemented in [5].

$$\varepsilon_{MS} = \frac{1}{N} \sum_{i \in J} \| \Phi(x_i) - \sum_{j \in J} \beta_{ij} \Phi(x_j) \|^2. \quad (2)$$

The GKPCA method, therefore only needs to determine the optimal subset J from I. As shown in [5], we can choose to minimize the upper bound where

$$\varepsilon_{MS} \le \frac{1}{N}(N-n) \sum_{i \in I \setminus J} \max \| \Phi(x_i) - \tilde{\Phi}(x_i) \|^2. \quad (3)$$

The task can be solved by an iterative greedy algorithm in [5]. The number of iterations of the algorithm equals to the number n of selected basis vectors. The number n can be equal to the number of all training vectors N at most. However, it is reasonable to stop the algorithm earlier. It is natural to stop the algorithm if the one of the conditions is satisfied: mean square reconstruction error ε_{MS} falls below prescribed limit; maximal error $\max \| \Phi(x_i) - \tilde{\Phi}(x_i) \|^2$ falls below prescribed limit; or the number of basis vectors achieves prescribed limit.

Further optimization of the GKPCA method can be found in [5]. Nevertheless, given the original training data X, it is possible to use the GKPCA method to find out X_S, a subset of X, which has similar linear span in the principal feature space.

In contrast to the ordinary KPCA method, the subset X_S does not contain all the training vectors. Training the KPCA method using the subset X_S will result in of an $n \times n$ size of a reduced kernel matrix K, and therefore reduction of the evaluation cost. The basis vectors can be selected by the GKPCA method which has low computational requirements and allows on-line processing of larger data sets.

III. Fuzzy C-means Algorithm

The Fuzzy C-Means (FCM) algorithm was introduced by J. C. Bezdek in [6]. The algorithm classifies the extracted feature data by the PCA, KPCA and GKPCA methods, and obtains their fuzzy partition matrix (class membership matrix) $U_{c \times n} = \{u_{ij}\}$ in the case. Each column of class membership matrix U is the distribution of the class belonging of its corresponding sample.

Suppose the extracted feature data are $X = \{x_1, x_2, \cdots, x_n\} \subset R^p$ by the PCA, KPCA and GKPCA methods, n is the number of samples, p is the number of principal components, and c is the number of clusters. If X is classified into c clusters, then the objective function of the FCM algorithm is defined in (4), and the fuzzy partition matrix $U_{c \times n} = \{u_{ij}\}$ of X is subjected to the conditions given in (5).

$$\min J(X, U, V) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m d_{ij}^2. \quad (4)$$

$$\begin{cases} \sum_{i=1}^{c} u_{ij} = 1, \ j = 1, \cdots, n \\ n > \sum_{j=1}^{n} u_{ij} > 0, \ i = 1, \cdots, c \\ 1 \ge u_{ij} \ge 0, \ i = 1, \cdots, c, \ j = 1, \cdots, n \end{cases}. \quad (5)$$

Where $U_{c \times n} = \{u_{ij}\}$ is the fuzzy partition matrix, and u_{ij} is the degree of membership that the sample x_j belongs to the cluster center v_i; $V_{c \times q} = \{v_i, i = 1, ..., c\}$ is the set of cluster centers, and v_i is the cluster center of class i; m is the weighting exponent that controls the fuzziness of the membership function; d_{ij} is a distance measure between the sample x_j and the cluster center v_i.

By utilizing Lagrange multipliers, the minimization of the objective function J in (4) is performed in subject to the restriction conditions in (5). The cluster centers V and the optimal membership matrix U are obtained by (6) and (7), respectively.

$$v_i = \sum_{j=1}^{n} u_{ij}^m x_j \Big/ \sum_{j=1}^{n} u_{ij}^m, \ i = 1, ..., c. \quad (6)$$

$$u_{ij} = 1 \Big/ \sum_{k=1}^{c} (d_{ij}/d_{kj})^{2/(m-1)}, \ i = 1, ..., c, \ j = 1, ..., n. \quad (7)$$

Where $d_{ij} = \| v_i - x_j \|, i = 1, ..., c, j = 1, ..., n$, it is chosen as Euclidean distance here.

IV. Experimentation and Results Analysis

A. Experimental data

Two examples are carried out to compare the performance of the PCA, KPCA and GKPCA methods using two data sets of Iris and Landsat Satellite in [7]. The descriptions of two data sets are shown in TAB. I.

TABLE I. THE DESCRIPTIONS OF TWO DATA SETS (NUMBER)

Data Sets	Samples	Classes	Attributes	Training Data	Test Data
Iris	150	3	4	90	60
Landsat Satellite	6435	6	4	4435	2000

B. Flowchart of the classification

Flowchart of the classification experiments is showed in Fig. 1. Given the data set $X = \{x_1, x_2, \cdots, x_n\} \subset R^p$, the data standardization method in (8) is used to obtain normalized data and balance the data impact on the results of the classification.

$$\tilde{x}_{jq} = (x_{jq} - \mu_q)/\sigma_q, \ \ j = 1, \cdots, n, \ \ q = 1, \cdots, p. \tag{8}$$

Where x_{jq} is the pth feature of the sample x_j, $\mu_q = \frac{1}{n}\sum_{j=1}^{n} x_{jq}$ is the mean vector, $\sigma_q^2 = \frac{1}{n}\sum_{j=1}^{n}(x_{jq} - \mu_q)^2$ is the variance vector of the qth feature vector, $q = 1, \cdots, p$.

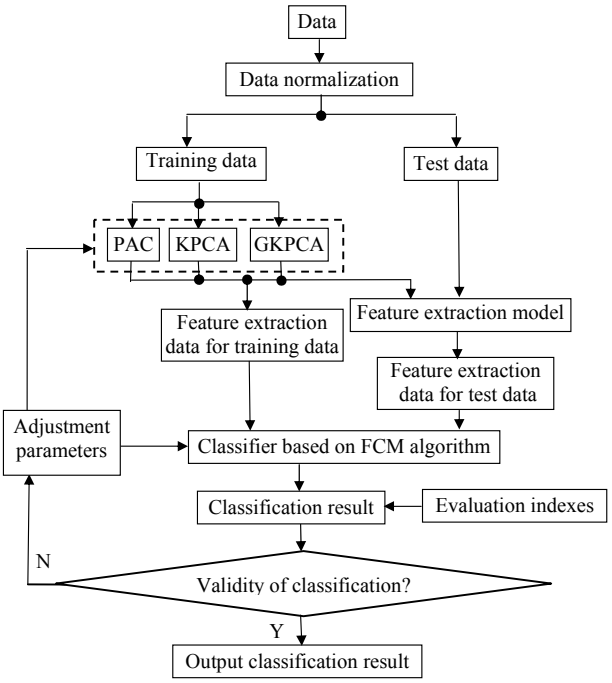

Figure 1. Flowchart of the classification.

C. Results of training data reduction by GKPCA method

The GKPCA method uses Radial Basis Functions (RBF) kernel in (9). In the GKPCA method, let the mean squared errors $\varepsilon_{MS} = 10^{-6}$, the desired maximal error $\varepsilon_{Max} = 10^{-6}$, and the value of kernel parameters σ is shown in TAB. II. Extracted subsets X_S from the training data X of Iris and Landsat Satellite by the greedy technology of the GKPCA method are also shown in TAB. II.

$$K(x_i, x_j) = \exp(-\| x_i - x_j \|^2 / (2\sigma^2)). \tag{9}$$

TABLE II. EXTRACTED SUBSETS FROM TRAINING DATA BY GKPCA

Data Sets	Kernel Parameters	Samples Number of Subset	Percentage of Training Data Reduction (%)
Iris	3	22	75.6
Landsat Satellite	8	783	82.3

D. Classification results and performance evaluations

The feature extraction data is classified by the FCM algorithm. In the FCM algorithm, set the weighting exponent $m = 2$ in [8], the convergent threshold $\varepsilon = 10^{-6}$, maximally iterative number $T_{\max} = 50$.

Clustering results are evaluated by classification accuracy and clustering validity indices, related to the inherent features of these extracted data. A few widely known validity indices, such as partition coefficient V_{pc}, partition entropy V_{pe} in [9] and Xie–Beni index V_{xb} in [10], which are chosen to evaluate partitions by the FCM algorithm from these extracted data by the PCA, KPCA and GKPCA methods. The partition coefficient V_{pc} and partition entropy V_{pe} in (10) and (11), respectively, introduced by Bezdek, are defined using the memberships from a classification algorithm. They indicate the degree to which a partition is unambiguous. The Xie–Beni index V_{xb} in (12) examines separation of with-class, and compactness of between-class.

$$V_{pc}(U, c) = \frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^2. \tag{10}$$

$$V_{pe}(U, c) = -\frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}\log_2(u_{ij}). \tag{11}$$

$$V_{xb}(U, v, c, X) = \frac{\sum_{i=1}^{c}\sum_{j=1}^{n}(u_{ij}^m \left\| X_j - v_i \right\|^2)}{N(\min\limits_{i,j=1,i\neq j}^{c} \left\| v_i - v_j \right\|)}. \tag{12}$$

Where $1/c \le V_{pc}(U, c) \le 1$, and $0 \le V_{pe}(U, c) \le \log_2 c$.

The best cluster is achieved when the value of the partition coefficient V_{pc} is high, and partition entropy V_{pe} and Xie-Beni index V_{xb} are low.

Classification performances of Iris data and Landsat Satellite data by several methods are shown in TAB. III and TAB. IV, respectively.

TABLE III. CLASSIFICATION PERFORMANCE OF IRIS

Data Sets	Methods	Recognition Rates (%)	Vpc	Vpe	Vxb
Training Data	FCM	93.3	0.8046	0.5194	0.2068
	PCA+FCM	95.0	0.8375	0.4339	0.1648
	KPCA+FCM	95.0	0.8634	0.3595	0.0263
	GKPCA+FCM	95.0	0.8634	0.3595	0.0297
Test Data	FCM	90.0	0.7849	0.5694	0.2473
	PCA+FCM	90.0	0.8191	0.4860	0.1982
	KPCA+FCM	90.0	0.8574	0.3865	0.0263
	GKPCA+FCM	90.0	0.8574	0.3865	0.0297

TABLE IV. CLASSIFICATION PERFORMANCE OF LANDSAT SATELLITE

Data Sets	Methods	Recognition Rates (%)	Vpc	Vpe	Vxb
Training Data	FCM	84.6	0.7792	0.5845	0.3382
	PCA+FCM	84.6	0.7704	0.5832	0.2924
	KPCA+FCM	85.6	0.7884	0.4785	0.0463
	GKPCA+FCM	85.2	0.7828	0.4834	0.0483
Test Data	FCM	84.4	0.7689	0.5975	0.3404
	PCA+FCM	84.5	0.7604	0.5792	0.2994
	KPCA+FCM	85.0	0.7981	0.4865	0.0563
	GKPCA+FCM	84.7	0.7868	0.4934	0.0583

According to simulation experiment over, the results show that the fuzziness of the partition is reduced by feature extraction, and the superiority of both the KPCA and GKPCA methods over the PCA method in feature extraction. Simulation results show both the KPCA and GKPCA methods are more superior to the PCA method in feature extraction. The GKPCA method will tend towards the KPCA method feature extraction as more percentage of training data is included in the reduced set, whilst the GKPCA method results in lower evaluation cost due to the reduced training set. The experiments show that the GKPCA method can significantly reduce the complexity of the found classifiers while retaining their accuracy.

V. CONCLUSIONS

The KPCA method, however, is that the storage of training data in terms of the dot products, is too expensive since the size of kernel matrix increases quadratically with the number of training data. So, a more efficient feature extraction method, the GKPCA method, is applied to reduce training data and nonlinear feature extraction in classification. The reduced set method aims to find a new kernel expansion and well approximates the original training data. Simulation results show both the KPCA and GKPCA methods are more superior to the PCA method in feature extraction. Feature extraction performance of the GKPCA method will tend towards one of the KPCA method, whilst the GKPCA method results in lower evaluation cost due to the reduced training set. The theoretical analysis and experimental results show the advantages of the GKPCA method in terms of computational efficiency, storage space, and nonlinear feature extraction capability, especially when the number of training data is large. In a word, the GKPCA method can significantly reduce the complexity while retaining their accuracy in classification.

ACKNOWLEDGMENT

The research is supported by Key Project of Sichuan Provincial Department of Education (No.11ZA124), Open Fund Project of Artificial Intelligence Key Laboratory of Sichuan Province (No. 2011RYJ02), and Talent Introduction Project of Sichuan University of Science and Engineering, (No. 2012RC21).

REFERENCES

[1] B. Schölkopf, A. Smola, and K. R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," Neural Computation, vol.10, pp.1299–1319, July 1998.

[2] B. Schölkopf and A. J. Smola, Learning with Kernels, Cambridge: MIT Press, 2002, pp.35–71.

[3] V. Franc and V. Hlaváč, Greedy algorithm for a training set reduction in the kernel methods, In N. Petkov and M. A. Westenberg, editors, Computer Analysis of Images and Patterns, Berlin: Springer-Verlag, 2003, pp.426–433.

[4] V. Franc, Optimization algorithms for kernel methods, PhD thesis, Prague: Czech Technical University, 2005, pp.87–103.

[5] T. Tangkuampien and D. Suter, "Human Motion De-noising via Greedy Kernel Principal Component Analysis Filtering," 18th International Conference on Pattern Recognition (ICPR 06) , Pattern Recognition, 2006, pp.457–460.

[6] J. C. Bezdek and W. F. Ehrlich, "FCM: The fuzzy c-mean clustering algorithm," Computers and Geoscienc., vol.10, February 1984, pp.191–203.

[7] A. Asuncion and D. J. Newman. UCI Machine learning repository. (2012-12-10). http://www. ics.uci.edu/~mlearn/MLRepository.html. Irvine, CA: University of California, School of Information and Computer Science, 2007.

[8] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," IEEE Trans,Fuzzy Systems, vol.3, August 1995, pp.370–373.

[9] J. C. Bezdek, "Cluster validity with fuzzy sets," Journal Cybernet, vol.3, February 1974, pp.58–73.

[10] X. Xie and G. Beni, "A validity measures for fuzzy clustering," IEEE Transactions on Pattern Analysis and Machine Intelligence. vol.13, August 1991, pp.841–747.