Subpattern-Based Principal Component Analysis

Songcan Chen [*] Yulian Zhu

Dept. of Computer Science and Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, People's Republic of China.

Abstract

We propose a subpattern-based principle component analysis (SpPCA). The traditional PCA operates directly on a whole pattern represented as a vector and acquires a set of projection vectors to extract global features from given training patterns. SpPCA operates instead directly on a set of partitioned subpatterns of the original pattern and acquires a set of projection sub-vectors for *each partition* to extract corresponding local sub-features and then synthesizes them into global features for subsequent classification. The experimental results show that the proposed SpPCA has (much) better classification performances on all the real-life benchmark datasets than PCA.

Keywords: Principle Component Analysis (PCA); Subpattern PCA (SpPCA); Feature extraction; Pattern recognition.

1. Introduction

The traditional PCA [1] is a very effective approach of extracting features and has successfully been applied in pattern recognition such as face classification [2]. It operates directly on whole patterns represented as (feature) vectors to extract so-needed global features for subsequent classification by a set of previously found global projectors from a given training pattern set, whose aim is to maximally preserve original pattern information after extracting features, *i.e.*, reducing dimensionality. In this paper, we develop another PCA operating directly on subpatterns rather than on whole pattern These subpatterns are formed via a

[*] Corresponding author: Tel: +86-25-489-1216; +86-25-489-3777.
E-mail: s.chen@nuaa.edu.cn (S.C. Chen), lianyi_1999@sohu.com (Y.L. Zhu).

partition for an original whole pattern and utilized to compose multiple training subpattern sets for the original training pattern set. In this way, SpPCA can independently be performed on individual training subpattern sets and finds corresponding local projection sub-vectors, and then uses them to extract local sub-features from any given pattern. Afterwards, these extracted sub-features from individual subpatterns are synthesized into a global feature of the original whole pattern for subsequent classification.

2. Proposed SpPCA

SpPCA includes two steps. In the first step, an original whole pattern denoted by a vector is partitioned into a set of equally-sized subpatterns in non-overlapping ways and then all those subpatterns sharing the same original feature components are respectively collected from the training set to compose corresponding training subpattern sets. Secondly, PCA is performed on each of such subpattern sets. More specifically, we are given a set of training patterns $X=\{X_1, X_2, \cdots, X_N\}$ with each column vector X_i (i=1, 2, … , N) having m dimensions. Now according to the first step, an original whole pattern is first partitioned into K d-dimensional subpatterns in a nonoverlapping way and reshaped into a d-by-K matrix $X_i=(X_{i1}, X_{i2}, \cdots, X_{iK})$, with $X_{ij}=(x_{i((j-1)d+1)}, \cdots, x_{i(jd)})^T$ being the jth subpattern of X_i and i=1,2,…, N and j=1,2,…,K. And then according to the second step, we construct PCA for the jth subpattern set $SP_j=\{X_{ij}, i=1,2,\cdots,N\}, j=1,2,\cdots,K$ to seek its projection vectors $\Phi_j=(\varphi_{j1}, \varphi_{j2}, \cdots, \varphi_{jl})$ based on minimizing the reconstructed (local) error (RCE) or maximizing the total scatter in the projected space. Analogous to the Eigenface method [2], we first define the jth total subpattern scatter (sub-scatter) matrix S_j as $S_j=\frac{1}{N}\sum_{i=1}^{N}(X_{ij}-\overline{X}_j)(X_{ij}-\overline{X}_j)^T$, where $\overline{X}_j=\frac{1}{N}\sum_{i=1}^{N}X_{ij}, j=1,2,\cdots,K$ are subpattern means. Here it is easy to prove that all total sub-scatter matrices are positive semi-definite and their scales are all dxd. And then find independently each set of projection sub-vectors by means of the following eigenvalue-eigenvector system under the constraints $\Phi_j^T \Phi_j = I_j, j = 1,2,\cdots,K$,

$$S_j \Phi_j = \Phi_j \Lambda_j \qquad (1)$$

where I_j is an identity matrix and Λ_j is a diagonal matrix composed by the first l largest non-negative eigenvalues of S_j in a descending order and thus their corresponding first l local eigenvectors compose the Φ_j. After obtaining all individual projection sub-vectors from the partitioned subpattern sets, we can extract corresponding sub-features Y_j from any subpattern of a given whole pattern $Z = (Z_1, Z_2, \cdots, Z_K)$ with Z_j being a subpattern with the dimension tallied with each corresponding partition in terms of (2):

$$Y_j = \Phi_j^T Z_j, j = 1,2,\cdots,K \qquad (2)$$

Then synthesize them into a global feature as follows:

$$Y = (Y_1^T, Y_2^T, \cdots, Y_K^T)^T = (Z_1^T \Phi_1, Z_2^T \Phi_2, \cdots, Z_K^T \Phi_K)^T \qquad (3)$$

It is not difficult to find that when K=1 and d=m, SpPCA reduces to the traditional PCA. In addition, the dimension of Y is still equal to that of projection vector found by PCA on the whole pattern. Now on the basis of the synthesized global features, we can use the nearest neighbor (NN) rule [3] to perform pattern classification.

3 Experimental Results

The experimental datasets are from publicly attainable 8 benchmark datasets including a derived one. For producing as many partitions as possible, all dimensions of patterns are not less than 12. Here we just give brief experimental conditions because of space limited.

1) ORL face data[1] (40 classes, 10 image patterns each class, 5 patterns each class for training and the rest

[1] AT&T Lab, Cambridge, http://www.cam-orl.co.uk/facedatabase.html;

for testing); 2) Letter data[2] (10 classes, 50 text patterns each class, 25 each class for training and the rest for testing); 3) Waveform data[3] (3 classes, 500 vector patterns each class, 250 each class for training and the rest for testing); 4) Water-treatment data[3] (2 classes, 65 and 51 vector patterns respectively, 25 each class for training and the rest for testing); 5)Wine data[3] (3 classes, 59, 71and 48 vector patterns respectively, 24 each class for training and the rest for testing); 6)Sonar data[4] (2 classes, 111 and 97 vector patterns respectively, 50 each class for training and the rest for testing); 7) Musk-Clean2[3] (2 classes, 1017 and 5581 vector patterns respectively, 500 each class for training and the rest for testing); 8) Musk-Clean2 (the same as 7) but the last 6 dimensions of each pattern are omitted (by us) mainly for generating more partitions).

The experimental results tabulated in Table 1 are averaged on 10 independent runs for each dataset. The results show that SpPCA has better classification accuracies (underlined and bolded) on *all* datasets used here than PCA, especially on the Wine dataset, where the increased accuracy attains 30%. Such increases on all the experimental datasets seem able to get partial interpretation from the viewpoint of decision function based on the NN rule. For PCA, its decision is made as follows: for any unknown whole pattern ***x***, if $i = \arg\min_{l}\{\left\|\Phi^T(x - x_l)\right\|^2$, then $\boldsymbol{x} \in$ class *i*. The decision can equivalently be changed to the corresponding decision functions $f_{PCA}(x) = [x - (x_l + x_i)/2]^T \Phi^T \Phi[x_l - x_i]$, where ***x****ᵢ* and ***x****ₗ* are the *i*th and *l*th whole training patterns respectively and Φ is the projection matrix found by PCA. For SpPCA, the corresponding decision functions are similarly obtained by $f_{SpPCA}(x) = \sum_{k=1}^{K}[x^{(k)} - (x_i^{(k)} + x_l^{(k)})/2]^T \Phi_k^T \Phi_k [x_l^{(k)} - x_i^{(k)}]$, where *K* and Φ_k are defined as before, and $x^{(k)}, x_i^{(k)}\ and\ \ x_l^{(k)}$ are the corresponding *k*th subpatterns of patterns ***x***, $\boldsymbol{x_i}$ and $\boldsymbol{x_l}$. From the decision functions of both, it is easy to observe that the PCA decision function is linear and while the

[2] Available on http://sun16.cecs.missouri.edu/pgader/ CECS477/NNdigits.zip
[3] UCI repository of machine learning databases, http://www.ics.uci.edu/~mlearn/MLRository.html.
[4] Available from CMU-AI Repository, ftp://ftp.cs.cmu.edu/afs/cs/project/connect/bench/.

SpPCA's is a mixture of *K* piece-wise linear functions in *d* dimensional space due to different Φ_k. Possibly it is a very point that makes SpPCA greatly boost PCA performances and attractive. Finally, when *K* is taken as 1, the decision functions of both are identical, which states again that PCA is indeed a special case of SpPCA.

Insert Table 1 here

4 Conclusions

We proposed a powerful new approach of extracting features based on PCA and pattern partitioning technique, and made PCA become a special case of the proposed SpPCA. By extracting local sub-features from partitioned subpattern sets and then synthesizing them into global features for subsequent classification. As a result, classification accuracies incorporating the NN rule on all datasets employed here are greatly improved.

Acknowledgements

We thank National Science Foundations of China and of Jiangsu under Grant Nos. 60271017 and BK2002092, the "QingLan" Project Foundation of Jiangsu Province and the Returnee Foundation of China Scholarship Council for partial supports respectively.

References

[1] T. Young, K-S Fu, Handbook of pattern recognition and image processing, Academic Press, 1986.

[2] M. Turk, A. Pentland, Eigenfaces for recognition, Journal Cognitive Neuroscience, 3(1)(1991)71-86.

[3] G. Loizou, S.J. Maybank, The nearest neighbor and the Bayes error rates, IEEE Trans. Patt. Anal. & Mach. Intell., vol.9 (1987)254-262,.

Table 1 Classification Accuracy (%) Comparison based on the NN rule (Cs-Classifiers)

Datasets	Cs	Accuracies	PNs
ORL	PCA	93.05 (112)	37
(112x92D)[1]	SpPCA	**95.45** (112)[2]	**54**[3]
Letters	PCA	88.89(24)	122
(24x18D)	SpPCA	**89.52**(24)	**17**
Waveform	PCA	77.36	18
(21D)	SpPCA	**78.79**(7)	**5**
Water	PCA	70.91	1
(38D)	SpPCA	**87.73**(19)	**18**
Wine	PCA	51.32	2
(12D)	SpPCA	**81.51** (6)	**6**
Sonar	PCA	78.06	16
(60D)	SpPCA	**82.69** (4)	**1**
Musk2	PCA	80.24	34
(166D)	SpPCA	**86.31** (83)	**12**
Musk2	PCA	79.98	32
(160D)	SpPCA	**86.58** (20)	**2**

1: The (D)imensions of original whole patterns.

2: In (*d*), *d* denotes the dimension of partitioned subpattern.

3: Projection numbers (PNs) of achieving corresponding accuracies.