

# An experimental study of Sub-Pattern based Principal Component Analysis and Cross-Subpattern-Correlation based Principal Component Analysis

Atul Negi<sup>1</sup>, Vijaya-Kumar Kadappa<sup>2</sup>

<sup>1</sup>Dept. of Computer and Information Sciences, University of Hyderabad, A.P., India  
Email:atulcs@uohyd.ernet.in, atul.negi@gmail.com

<sup>2</sup>Dept. of Computer Applications, Vasavi College of Engineering, Hyderabad, India  
Email:kvkumar@vce.ac.in, kvkumar28@yahoo.com

## Abstract

Principal Component Analysis (PCA) is a well established technique for dimensionality reduction. However, the time complexity of classical PCA and classification accuracy have been the focus of much research. One of such approaches is 'Sub-Pattern based Principal Component Analysis (SubPCA)', where SubPCA reduces the time complexity by a regular partitioning of feature space. In our work, we examine certain aspects of SubPCA especially the 'cross-subpattern correlation'. SubPCA considers only local correlations, but has no mechanism to bring out 'global correlations' across sub-patterns in the data. In this paper, we address the cross-subpattern correlation issue by proposing an interesting variation of the PCA technique which extracts such correlations aiding in dimensionality reduction. We demonstrate our technique by application upon 5 UCI repository databases and show the superiority of the proposed method in comparison to the SubPCA and PCA techniques.

**Keywords:** Principal component analysis, subpattern based PCA (SubPCA), cross-subpattern correlation

## 1 Introduction

Principal Component Analysis (PCA) is one of the widely used techniques for dimensionality reduction. A useful discussion on PCA may be found in any of the popular books [1][2][3][4]. It's many applications include data compression, visualisation, pattern recognition, exploratory data analysis, etc.,

We shall use the notation as described below. The set,  $\mathbf{X}$ , of all original patterns is denoted by  $\mathbf{X} = [(\mathbf{X}_1, \theta_1), (\mathbf{X}_2, \theta_2), \dots, (\mathbf{X}_N, \theta_N)]$  where  $N$  is the cardinality of  $\mathbf{X}$ . Each original pattern,  $\mathbf{X}_i$ , is a vector of dimension  $d$  which is given by  $\mathbf{X}_i = (x_{i_1}, x_{i_2}, \dots, x_{i_d})$  where  $x_{i_1}, x_{i_2}, \dots, x_{i_d}$  are feature values that represent  $\mathbf{X}_i$ . Every pattern  $\mathbf{X}_i$  is associated with a class label  $\theta_i$ , where  $\theta_i \in C, \forall i = 1, 2, \dots, N$ . Here,  $C = \{c_1, c_2, \dots, c_n\}$  is the possible collection of class labels,  $n$  is the number of classes under consideration.  $\mathbf{SP}_j$  is the  $j^{th}$  sub-pattern set, which is represented by  $\mathbf{SP}_j = (x_{1_{d_1(j-1)+1}}, x_{1_{d_1(j-1)+2}}, \dots, x_{1_{d_1(j-1)+d_1}}, \dots, x_{N_{d_1(j-1)+1}}, x_{N_{d_1(j-1)+2}}, \dots, x_{N_{d_1(j-1)+d_1}})$ . That is,  $\mathbf{SP}_j$  is the collection of  $j^{th}$  subpatterns of all original patterns  $\mathbf{X}$ , where  $j = 1, 2, \dots, k$  and  $k$  is number of equally-sized subpatterns. A  $j^{th}$  subpattern of original pattern  $\mathbf{X}_i$  is represented

by  $\mathbf{Y}_j^i = (x_{i_{d_1(j-1)+1}}, x_{i_{d_1(j-1)+2}}, \dots, x_{i_{d_1(j-1)+d_1}})$ , where  $d_1$  is the size of the subpattern.  $d_2$  is the size of last sub-pattern, if  $d$  is not a multiple of  $d_1$ .  $\mathbf{Y}_j^e$  is the set of  $r$  PCs extracted from  $\mathbf{SP}_j$ .  $\mathbf{Z}_{N \times kr}$  is the reduced pattern data matrix obtained from SubPCA method.  $\mathbf{Z}_{N \times r}^X$  is the final reduced data obtained from SubXPCA method, where  $r^X$  is the number of PCs extracted from  $\mathbf{Z}$ .

PCA is concerned with explaining variance-covariance structure through a few linear combinations of the original variables (features). Although  $d$  components are required to reproduce the total data ( $\mathbf{X}$ ) variability, often much of this variability can be accounted for by a small number,  $m$  ( $< d$ ), of the principal components (PCs). If so, there is (almost) as much information in the  $m$  PCs as there is in the original  $d$  variables. Thus  $m$  PCs can then be used to approximate the original data set, by reducing to one consisting of  $N$  measurements on  $m$  PCs (new variables/features). The PCs are those uncorrelated linear combinations whose variances (given by eigenvalues) are as large as possible. The first PC is the linear combination ( $\mathbf{e}_1^t \mathbf{X}$ ) with maximum variance in the direction of eigenvector ( $\mathbf{e}_1$ ) corresponding to first highest eigenvalue ( $\lambda_1$ ). Similarly, second PC is in the direction of ( $\mathbf{e}_2$ )

corresponds to second highest eigenvalue ( $\lambda_2$ ) and so on. Any two eigenvectors are in orthogonal directions [4]. The usefulness and hence popularity of PCA comes from its properties – it is an optimal linear scheme (in terms of mean squared error) for reducing data to a lower dimensionality and uses only matrix multiplication operations for reduction and reconstruction.

However, classical PCA suffers from large time complexity ( $O(Nd^2)$ ) just to calculate covariance matrix for high dimensional data. To overcome this problem, sub-pattern based PCA (SubPCA) technique [5] was proposed. The authors there demonstrated a better classification accuracy. SubPCA partitions the original pattern data into  $k$  number of equally-sized subpatterns. The set of  $j^{th}$  subpatterns of all original patterns constitutes the  $j^{th}$  subpattern set,  $\mathbf{SP}_j$ . Classical PCA is performed on each of  $\mathbf{SP}_j$  ( $j = 1, \dots, k$ ), to extract  $r$  local features from each  $\mathbf{SP}_j$  and they are used to synthesize a global feature set of original representative of the data.

In this paper, we investigate several aspects of SubPCA and bring up various issues, viz – feature order dependency, selection of subpattern size, truncation/padding of features, choice of principal components, sensitivity to missing & outlier data and cross-subpattern correlation (See sec 3). Here we present a solution to the issue ‘cross-subpattern correlation’ only (See sec 4).

It is evident that SubPCA considers only local correlation structures (i.e. local to  $\mathbf{SP}_j$ ). This technique is very useful, if such high-correlations are present and no correlations exist between features spread across different subpattern sets ( $\mathbf{SP}_j$ ). SubPCA does not consider correlations which span across subpattern sets (See sec 3.6). Such correlations may be useful for dimensionality reduction and better classification. In this context, we propose a novel technique, where (i) SubPCA is applied to get local feature data from subpattern sets and (ii) classical PCA is performed to get final reduced data. In a nutshell, the proposed technique reduces the dimensionality of the data already reduced by SubPCA and increases classification accuracy as well (see sec 4 for details).

In section 2, we review the salient aspects of SubPCA [5]. Some fundamental issues in the context of SubPCA are presented in section 3. In section 4, cross-subpattern correlation based PCA (SubXPCA) technique is proposed. Experimental results on several benchmark databases (UCI repositories) and time complexities of SubPCA, SubXPCA are discussed in section 5. We conclude with some comments in section 6.

## 2 Sub-pattern based PCA

We review briefly the SubPCA technique and a detailed discussion may be found in [5].

### SubPCA Algorithm

**Step 1:** The original pattern data,  $\mathbf{X}$ , is partitioned into suitable number ( $k$ ) of equally-sized subpatterns without common feature(s) between any two subpatterns. The set of  $j^{th}$  subpatterns of all original patterns constitutes  $j^{th}$  subpattern set,  $\mathbf{SP}_j$ .

**Step 2:** Classical PCA is performed on each of  $k$  subpattern sets ( $\mathbf{SP}_j$ ) to extract  $r$  local features to get combined local-features set ( $\mathbf{Z}$ ) of original pattern set,  $\mathbf{X}$ .

## 3 Basic issues with SubPCA

SubPCA is very interesting, but as presented in [5], it leaves several issues to be explored. In the following, we discuss some issues.

### 3.1 Feature-order dependency

SubPCA selects  $d_1$  features contiguously, in their order of appearance to form subpatterns, while dividing the given pattern into  $k$  subpatterns. The question arises, whether SubPCA produces results, which are independent of feature ordering. We investigated this by considering 20 random feature orders on 3 UCI benchmark databases (See sec 5) and the results are summarized in Table 1.

Table 1: Mean and variance of classification accuracies (using 1NN rule) with 20 randomized orders of features for SubPCA.

Data set	Mean	Variance
Waveform data	79	0.1
Musk data	84	0.2
Wine data	80.6	1.3

Using the results in table 1, one may suspect that the SubPCA is order dependent. To confirm whether SubPCA is order dependent or not, probably a sound statistical analysis may be required.

### 3.2 Selection of sub-pattern size

SubPCA is silent about how subpattern size selection should be done. All subpatterns have the same fixed arbitrary size. Therefore SubPCA approximates classical PCA, as subpattern size increases: if the number of subpatterns is one, both SubPCA and PCA are identical. If the subpattern size is too small, the SubPCA technique can not capture the correlation structures appropriately, thus poor

results may be obtained. Hence, proper judgement is required to select appropriate subpattern size. One could use a clustering technique to group similar features [6]. The disadvantage would be that clustering will require an additional cost to obtain feature groups.

### 3.3 Truncation/Padding of features

Normally, if the pattern size  $d$  is not a multiple of subpattern size ( $d_1$ ), the size of the last subpattern ( $d_2$ ) is different from the size of previous subpatterns. While selecting subpattern size (See sec 3.2), one may consider, whether features in the last subpattern are to be truncated or some dummy features are to be added to do a padding-up to the size of previous subpatterns. SubPCA uses the option of truncating  $d_2$  features (if any) in the last subpattern. This option may severely affect the results, if the truncated features have vital discriminative information. The alternative strategy is to pad dummy features which do not contain any discriminative information in such a way that the last subpattern size equals to  $d_1$ . Another strategy is not to pad the last subpattern, instead handle it separately with a different subpattern size ( $d_2$ ).

### 3.4 Selection of principal components

The first few projection vectors (PVs) correspond to the highest eigenvalues. Hence these are used to project the given pattern data to get the principal components (PCs) for each pattern. The projected set of features is the set of new features for the original pattern. SubPCA also obtains these but, subpattern set-wise,  $SP_j$ . Several researchers are studying the problem of suitable PV selection (hence PCs) so as to capture maximum structure (variation) among the patterns. One useful article can be found in [7]. SubPCA selects same number ( $r$ ) of PVs (thus  $r$  local features) from each  $SP_j$  and selects  $kr$  number of PCs/PVs in total. Another strategy of choosing PVs is, to fix some threshold ( $t$ ) and use it to select PVs. In other words, if  $(\lambda_i/\sum\lambda_i) \geq t$  choose PV,  $e_i$ , corresponding to eigenvalue  $\lambda_i$ .

### 3.5 Sensitivity to missing data and outliers

PCA is highly sensitive to missing feature values in the data and outlier data. SubPCA suffers from the same problem. A useful approach to tackle this can be found in [8].

### 3.6 Cross-subpattern correlations

As stated, SubPCA forms a combined feature-set ( $Z$ ) from the original pattern set, after extracting local features from each  $SP_j$ . Here local features are extracted based on correlation structure present in the subpattern set ( $SP_j$ ) only. The features thus extracted from different  $SP_j$ s, may be correlated. We call such correlation as a ‘cross-subpattern correlation’. SubPCA does not consider cross-subpattern correlations.

We address this issue in this paper by proposing a new technique based on cross-subpattern correlations (See sec 4). Few cross-subpattern correlations are illustrated in figures 1, 2, for the data sets of UCI repositories (See sec 5). These figures are drawn as per the procedure described below.

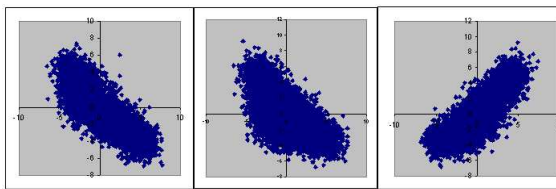


Figure 1: Some cross-subpattern correlations in waveform data (See sec 5)

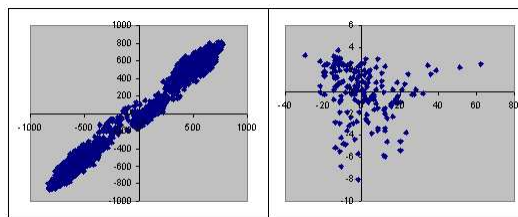


Figure 2: Some cross-subpattern correlations in musk and wine data (See sec 5)

1. **Cross-subpattern correlations in waveform data:** We selected first PC from 3 subpattern sets ( $SP_j; j = 1, 2, 3$ ). The graphs plotted between first PCs which are selected from (i)  $SP_1$ & $SP_2$  (ii)  $SP_1$ & $SP_3$  (iii)  $SP_2$ & $SP_3$  are shown in figure 1.
2. **Cross-subpattern correlations in musk and wine data:** We selected first PC from 2 subpattern sets ( $SP_j; j = 1, 2$ ). The graphs plotted between first PCs which are selected from respective  $SP_1$ & $SP_2$  for wine and musk data respectively are shown in figure 2.

Thus, we observe that the possibility of cross-subpattern correlations allows for further dimensionality reduction.

Table 2: Classification accuracies of various data sets based on 1NN rule:

Data set	PVs per SP	Total no. of PVs			Accuracy <sup>+</sup>			Executn. time (secs.)		
		SPC	SXP	PCA	SPC	SXP	PCA	SPC	SXP	PCA
Waveform (21) <sup>a</sup> (7) <sup>b</sup>	5	15	2	2	78.3 (0.7)	82.1 (0.2)	82 (0.2)	10.73	2.11	2.15
Musk data (166) <sup>a</sup> (83) <sup>b</sup>	12	24	10	10	83.6 (1.2)	84.3 (1.5)	84.3 (1.3)	36.15	20.50	27.26
Musk data (166) <sup>a</sup> (55) <sup>b</sup>	8	24	9	9	84 (0.94)	84.7 (0.9)	83.5 (1.1)	33.73	16.93	26.03
Wine data (13) <sup>a</sup> (6) <sup>b</sup>	6	12	11	11	80.7 (13.2)	80.7 (13.2)	80.7 (13.2)	0.02	0.02	0.03
Breast cancer (30) <sup>a</sup> (10) <sup>b</sup>	4	12	4	*	89.3 (3.5)	89.3 (3.5)	*	0.11	0.06	*
Forest data (54) <sup>a</sup> (13) <sup>b</sup>	8	32	8	*	73.1 (0.4)	73.1 (0.4)	*	80.02	22.17	*

<sup>a</sup> The dimension of the original pattern ; <sup>b</sup> The dimension of the subpattern

\* : The tqli function [9] for computing eigenvalues and eigenvectors took too many iterations

PVs per SP: No. of Projection Vectors per subpattern set in SubPCA; SPC: SubPCA; SXP: SubXPCA

+:The Mean and variance (variance is in parentheses) of 10 classification accuracies (See sec 5)

#### 4 Cross-subpattern-correlation based PCA (SubXPCA)

In this section, we propose the technique, SubX-PCA, which is based on SubPCA (Ref sec 2) and cross-subpatten correlation (Ref sec 3.6) in detail.

##### 4.1 Cross-subpattern-correlation based PCA algorithm:

In the follwing steps, we use the indices as follows.  $1 \leq i \leq N; 1 \leq j \leq k; 1 \leq p \leq d_1; 1 \leq s \leq kr$

1. Apply SubPCA (Ref sec 2) to the set of mean-corrected data  $\mathbf{X}$ .

1.1. Partition datum,  $\mathbf{X}_i$  into  $k$  equally-sized subpatterns.  $\mathbf{SP}_j$  is the set of  $j^{th}$  subpatterns of  $\mathbf{X}_i, \forall i$ .

1.2. For every  $\mathbf{SP}_j$ , repeat the following steps 1.2.1 to 1.2.4.

1.2.1. Compute covariance matrix,  $(\mathbf{C}_j)_{d_1 \times d_1}$ .

1.2.2. Compute eigenvalues  $(\lambda_p^j)$  and corresponding eigenvectors  $(\mathbf{e}_p^j)$ .

1.2.3. Select  $r (< d_1)$  eigenvectors corresponding to first  $r$  largest eigenvalues obtained in step 1.2.2. Let  $\mathbf{E}_j$  be the set of  $r$  eigenvectors (column vectors) selected in this step.

1.2.4. Extract  $r$  local features (PCs) from  $\mathbf{SP}_j$  by projecting  $\mathbf{SP}_j$  onto  $(\mathbf{E}_j)_{d_1 \times r}$  as follows. Let  $\mathbf{Y}_j^e$  be the reduced data in this step.

$$\mathbf{Y}_j^e = (\mathbf{SP}_j)(\mathbf{E}_j)$$

1.3. Combine  $\mathbf{Y}_j^e, \forall j$ , as shown below. Let  $\mathbf{Z}$  denotes such combined data.

$\mathbf{Z}_i = (y_1^e(i, 1), y_1^e(i, 2), \dots, y_1^e(i, r) \dots \dots \dots y_k^e(i, 1), y_k^e(i, 2), \dots, y_k^e(i, r))$  where  $\mathbf{Z}_i$  is the  $i^{th}$  row of  $\mathbf{Z}$ , which corresponds to  $\mathbf{X}_i$ .  $(y_j^e(i, 1), y_j^e(i, 2), \dots, y_j^e(i, r))$  is  $i^{th}$  row of  $\mathbf{Y}_j^e$ .

2. Apply classical PCA on  $\mathbf{Z}$  obtained in step 1.3.

2.1. Compute covariance matrix,  $(\mathbf{C}^X)_{(kr) \times (kr)}$  for the data  $\mathbf{Z}$ .

2.2. Compute eigenvalues  $(\lambda_s^X)$  and corresponding eigenvectors  $(\mathbf{e}_s^X)$ .

2.3. Select  $r^X (< kr)$  eigenvectors corresponding to first  $r^X$  highest eigenvalues obtained in step 2.2. Let  $\mathbf{E}^X$  be the set of  $r^X$  eigenvectors selected in this step.

2.4. Extract  $r^X$  global features (PCs) by projecting  $\mathbf{Z}$  (obtained in step 1.3) onto  $\mathbf{E}^X$ . Let  $\mathbf{Z}^X$  be the data obtained after projection in this step.

$$\mathbf{Z}^X = (\mathbf{Z})(\mathbf{E}^X)$$

2.5.  $\mathbf{Z}^X$  is the final reduced pattern data (corresponds to original data,  $\mathbf{X}$ ), which can be used for subsequent classification, recognition, etc.

##### 4.2 Discussion of SubXPCA:

In the first step, SubPCA technique is applied on the original pattern data, to reduce the time complexity as well as to take advantage of structure local to subpattern sets (which may improve the classification accuracy) as specified in [5].

In the second step, cross-subpattern correlations between features belong to  $\mathbf{Z}$  are found by computing covariance matrix ( $\mathbf{C}^{\mathbf{X}}$ ) of those features. This covariance matrix is used by classical PCA to reduce the dimensionality of  $\mathbf{Z}$ .

**Time complexity of SubXPCA:** In Covariance based PCA techniques, a large amount of time is spent in calculating covariance matrix. Hence, we concentrate only on time complexity of computation of covariance matrix.

Time complexity to calculate  $k$  covariance matrices in SubPCA is given by

$$T(SPCACov) = O(kN(d_1)^2) \quad (1)$$

Obviously, it is less than the time complexity of classical PCA technique ( $O(Nd^2)$ ).

Time complexity to calculate all covariance matrices in SubXPCA is given by

$$T(SXPACov) = O(kN(d_1)^2) + O(N(kr)^2) \quad (2)$$

### 4.3 Why is SubXPCA better than SubPCA?

SubXPCA computes correlations between local features belonging to different subpatterns. Using such cross-subpattern correlations SubXPCA summarizes the structure (variance) among the patterns in few PVs corresponding to first largest eigenvalues. In the best case, where high cross-subpattern correlations structures exist, very few eigenvectors (possibly one) are sufficient to abstract most of the structure. In the worst case, where poor correlations (almost no cross-subpattern correlations) exist, most of the eigenvectors (possibly all) may be required to abstract most of the structure in the data. SubPCA becomes a special case of SubXPCA, if there is zero cross-subpattern correlation. As explained in [10], PCA is able to retain meaningful information (variance) in the early axes (PCs/PVs corresponding to highest eigenvalues), where as variance associated to experimental error, measurement inaccuracy, and/or rounding (i.e. noise) is summarised in later axes (PCs/PVs corresponding to small eigenvalues).

SubPCA considers all  $kr$  local features, which may also include less-expressive and noisy features. SubXPCA summarizes the structure (variance) among these  $kr$  local features by the application of cross-subpattern concept and selects few salient features ( $r^{\mathbf{X}}$ ) which may increase the classification accuracy in addition to reduction of dimensionality.

## 5 Experimental results and analysis

We considered 5 publicly available databases from UCI repository of Machine Learning [11] for our experiments. (1) Waveform data (21 features, 3 classes with labels (0,1,2), 5000 Patterns, 250 patterns each class, for training, rest of them for testing.). (2) Musk data (166 features, 2 classes with labels (0,1), 6598 patterns, 500 patterns each class, for training, rest of them for testing). (3) Wine data (13 features, 3 classes with labels (1,2,3), 178 patterns, 24 patterns each class, for training, rest of them for testing). (4) Breast cancer (wdbc) data (30 features, 2 classes with labels (1 replaced for M, 2 replaced for B), 569 Patterns, 30 patterns each class, for training, rest of them for testing.). (5) Forest data (54 features, 7 classes with labels (1,2,3,4,5,6,7), 7000 Patterns (extracted 1000 per class from 581012 patterns), 400 patterns each class, for training, rest of them for testing.).

We used 1NN method for classification of the data. We generated 10 independent sets of training data for each of the data sets. For this training data, mean and variances of classification obtained are tabulated in table 2. We used Pentium 4 based system with system clock speed of  $10^6$  ticks/sec, to obtain the results listed in tables 1, 2, 3.

### 5.1 Dimensionality reduction by SubXPCA

A glimpse of dimensionality reduction by SubXPCA on 4 UCI databases is given in columns 5 & 6 of table 3. As shown in table 3, SubXPCA summarizes the most of the variation (in some cases it is almost equal to 100%) in the combined local-feature set ( $\mathbf{Z}$ ) of the 3 data sets. This is achieved by using correlations between features across subpatterns.

### 5.2 Performance of SubXPCA over SubPCA and PCA

By observing equations (1) and (2) (See sec 4.2), one would say that SubXPCA consumes more time than SubPCA just to calculate covariance matrices. However, SubXPCA uses fewer PVs than SubPCA (columns 3 & 4 in table 2 and columns 3 & 5 in table 3), and shows better classification accuracies than SubPCA as well (columns 6 & 7 in table 2). By using less number of PVs, SubXPCA saves computational costs to the largest extent in subsequent usage of reduced data (i.e. classification, transmission, etc). Hence SubXPCA shows superiority over SubPCA in terms of dimensionality reduction, time complexity and classification accuracy. In a nutshell, by observing

Table 3: Summarization of variance of PCs

Data set	PVs per SP	SubPCA		SubXPCA		PCA	
		Total PVs	Prop.	Total PVs	Prop.	Total PVs	Prop.
Waveform (21) <sup>a</sup> (7) <sup>b</sup>	5	15	82.4	2	70.4	2	62.2
				5	78	5	69
				10	89.2	10	79.1
						15	89
Musk data (166) <sup>a</sup> (83) <sup>b</sup>	12	24	83.2	2	66	2	55
				5	81	5	67.3
				10	94.2	10	79
				13	98	13	83
						24	92
Wine data (13) <sup>a</sup> (6) <sup>b</sup>	6	12	100	2	96.1	2	96.1
				5	99.7	5	99.7
Forest data (54) <sup>a</sup> (13) <sup>b</sup>	8	32	100	2	95	*	*
				5	99.9	*	*

Prop: Proportion of eigenvalues to the sum of all eigenvalues

\* : same as \* specified in table 2 ; PVs per SP: Same as table 2

(i) execution time (which includes computation of covariance matrices, finding eigenvalues and eigenvectors, INN classification time, etc), (ii) classification accuracies (both specified in table 2), and (iii) dimensionality reduction (specified in table 3), one can understand the superior performance of SubXPCA over SubPCA and PCA.

## 6 Conclusions

We have studied several fundamental issues to be addressed in the context of SubPCA for better dimensionality reduction and classification. The issue of ‘cross-subpattern correlation’, is used to propose a novel SubXPCA technique which reduces the dimensionality, improves classification accuracy and thus saving computational/processing costs in subsequent usage of the data. By our analysis we show that SubPCA is reduced to a special case of SubXPCA, in the worst case of SubXPCA, where cross-subpattern correlation is zero.

## References

- [1] I.T.Jolliffe, *Principal Component Analysis*. New York: Spinger-Verlag, 2002.
- [2] P.A.Devijver and J.Kitler, *Pattern recognition: A statistical approach*. London: Prentice Hall, 1982.
- [3] Duda and Hart, *Pattern classification*. John wiley & sons, 2004.
- [4] R.A.Johnson and D.W.Wichern, *Applied Multivariate Statistical Analysis, Third edition*. Prentice-Hall of India, pp. 356-394, 2001.
- [5] S.Chen and Y.Zhu, “Sub-pattern based principal component analysis,” *Pattern Recognition*, vol. 37, pp. 1081–1083, 2004.
- [6] K.M.Branson and S. Agarwal, “Structural principal component analysis,” *UCSD Technical Report*, October, 2003.
- [7] P.R.Peres-Neto and D.A.Kackson, “How many principal components ? stopping rules for determining the number of non-trivial axes revisited,” *Computational Statistics and Data Analysis, Elsevier*, 2005.
- [8] H. Chen, “Principal component analysis with missing data and outliers,” [www.caip.rutgers.edu/rriul/research/tutorials/tutorialrpca.pdf](http://www.caip.rutgers.edu/rriul/research/tutorials/tutorialrpca.pdf), visited on August 1st, 2005.
- [9] W.H.Press and S.A.Teukolsky, *Numerical recipes in C, Second edition*. Cambridge University Press, pp. 469-481, 2002.
- [10] Jr.H.G.Gauch, “Noise reduction by eigenvector ordination,” *Ecology*, vol. 63, pp. 1643–1649, 1982.
- [11] “UCI repository of machine learning databases,” [www.ics.uci.edu/~mllearn/MLRepository.html](http://www.ics.uci.edu/~mllearn/MLRepository.html), visited on August 1st, 2005.