Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition 41 (2008) 1398–1409
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/pr

SubXPCA and a generalized feature partitioning approach to principal component analysis

Kadappagari Vijaya Kumar[a,b,1], Atul Negi[a,*]

[a]*Department of Computer and Information Sciences, University of Hyderabad, Hyderabad 500046, India*
[b]*Department of Computer Applications, Vasavi College of Engineering, Hyderabad 500031, India*

Received 17 August 2006; received in revised form 20 July 2007; accepted 14 August 2007

Abstract

In this paper we propose a general feature partitioning framework to PCA computation and raise issues of cross-sub-pattern correlation, feature ordering dependence, selection of sub-pattern size, overlap of sub-patterns and selection of principal components. These issues are critical to the design and performance of feature partitioning approaches to PCA computation. We show several open issues and present a novel algorithm, SubXPCA which proposes a solution to the cross-sub-pattern correlation issue in the feature partitioning framework. SubXPCA is shown to be a general technique since we derive PCA and SubPCA as special cases of SubXPCA. We show SubXPCA has theoretically better time complexity as compared to PCA. Comprehensive experimentation on UCI repository data and face data sets (ORL, CMU, Yale) confirms the superiority of SubXPCA with better classification accuracy. SubXPCA not only has better time performance but is also superior in its summarization of variance as compared to SubPCA. SubXPCA is shown to be robust in its performance with respect to feature ordering and overlapped sub-patterns.

Keywords: Dimensionality reduction; Principal component analysis; Sub-pattern based PCA; Feature partitioning

1. Introduction

Principal component analysis (PCA) is one of the widely used techniques for dimensionality reduction with widespread applications to pattern recognition, exploratory data analysis, etc., [1–4]. PCA is concerned with summarizing the *variance–covariance structure* using a few linear combinations of the original set of d variables (features). Although d features are required to reproduce the total data variability, often much of this variability can be accounted for by a small number, m ($m < d$), of the principal components (PCs). To approximate the original data, a reduced set of m PCs is used. Those m are the uncorrelated linear combinations with the first m largest variances given by corresponding eigenvalues [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 the covariance matrix for high dimensional data. Reduction of time complexity is essential especially for the algorithms, where PCA is used fundamentally and is computed several times, for example, clusters of correlation connected objects [5]. Some of the approaches which aim to reduce time in computing PCs are random projection (RP) [6,7]. Degalla et al. demonstrated that fewer PCs are sufficient to produce a high accuracy by PCA, whereas RP needs a larger number of features to achieve good accuracy [6]. Fradkin and Madigan [7] found in their study that RP is inferior to PCA for dimensionality reduction. Thus although computation is speeded up by these approaches their performance does not compare well with PCA when it comes to dimensionality reduction.

* Corresponding author. Tel.: +91 40 23134031; fax: +91 40 23010780.
E-mail addresses: atulcs@uohyd.ernet.in, atul.negi@gmail.com (K. Vijaya Kumar), kvkumar28@yahoo.com, kadappakumar@gmail.com (A. Negi).
[1] Kadappagari is family name.

0031-3203/$30.00 © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2007.08.006

Speeding up computation of PCs is not the only point when studying advanced PCA computation methods. Additionally we need to see how to balance local and global feature properties while computing PCs. One approach to PCA that selects features locally is seen in the SubPCA algorithm [8], where features were partitioned into 'sub-patterns'. This approach showed good results on UCI and other data sets [8]. Further work using the sub-pattern concept with some modifications was presented as AwSubPCA [9] and sub-intra-personal subspace analysis (SISA) methods [10].

In this paper, we aim to provide a generalization from sub-patterns to a feature partitioning framework and bring up the important issues of cross-correlations across sub-patterns, overlapping of sub-patterns across features, feature order dependency, truncation/padding up of features, etc. We propose a feature partitioning based approach to principal component computation, called as SubXPCA. SubXPCA is a generalization since we show that SubPCA and PCA can be both derived as special cases of SubXPCA. SubXPCA balances the global PCs computation of PCA against the local viewpoint of SubPCA. We also prove the computational superiority of SubXPCA over PCA. Comprehensive experimentation shows the superiority of SubXPCA on UCI repository data, the well known Yale[11], CMU [12] and ORL[13] face data sets.

In the next section, we take up the major issues of feature partitioning as mentioned above. This is followed by a formal presentation of the SubXPCA algorithm. We show experimental results in Section 4 followed by concluding remarks in Section 5.

2. A generalized feature partitioning approach and PCA

In this section we explain the concept of a feature partitioning framework to PCA computation. We bring out the various feature partitioning issues such as cross-correlations across sub-patterns, feature order dependency, truncation/padding up of features and selection of PCs.

We shall use the notation as described below. The set, $\mathbf{X}$, of all original patterns is denoted by $(\mathbf{X})_{\mathbf{N}\times\mathbf{d}} = [(\mathbf{X_1}, \theta_1), (\mathbf{X_2}, \theta_2), \ldots\ldots, (\mathbf{X_N}, \theta_N)]$ where N is the cardinality of $\mathbf{X}$. Each original pattern, $\mathbf{X_i}$, is a vector of dimension d which is given by $\mathbf{X_i}=(x_{i_1}, x_{i_2}, \ldots, x_{i_d})$ where $x_{i_1}, x_{i_2}, \ldots, x_{i_d}$ are feature values that represent $\mathbf{X_i}$. Every pattern $\mathbf{X_i}$ is associated with a class label θ_i, where $\theta_i \in G, \forall i = 1, 2, \ldots, N$. Here, $G = \{g_1, g_2, \ldots, g_n\}$ is the possible collection of class labels, n is the number of classes under consideration.

2.1. Essential concepts

Following the notation explained previously, consider the original set of patterns, $\mathbf{X}$. Normally while doing PCA, we compute the covariance matrix ($\mathbf{C}$) for $\mathbf{X}$, then we find eigenvectors and eigenvalues. Instead in the feature partitioning approach, each instance $\mathbf{X_i}$ is divided into k sub-patterns, $\mathbf{P_i^j}$, $j = 1 \ldots k$. We let $\mathbf{SP_j}$ denote the set of jth sub-patterns from each of the original patterns. Now given jth sub-pattern set, $\mathbf{SP_j}$, we can proceed to extract features on this sub-pattern set by classical PCA procedure. This process is to be repeated for every sub-pattern set, $\mathbf{SP_j}$, $j = 1 \ldots k$. Finally the local features thus extracted are collated to form combined feature set ($\mathbf{Z}$).

2.2. Feature partitioning issues

2.2.1. Cross-sub-pattern correlations

The features extracted from different sub-pattern sets, $\mathbf{SP_j}$; $j = 1, \ldots, k$, may be correlated. We call such correlations as 'cross-sub-pattern correlations'. A good feature partitioning approach will not neglect such cross-sub-pattern correlations. These cross correlations help very much in dimensionality reduction. A few cross-sub-pattern correlations are illustrated in Figs. 1 and 2, for the data sets of UCI repositories. First, let us study the cross-sub-pattern correlations in waveform data (Fig. 1). Here the first PC from each of 3 sub-pattern sets ($\mathbf{SP_j}$ of waveform training data; $j=1, 2, 3$) as shown in Fig. 1(a,b,c). As shown in Fig. 1(a) almost entire variance of selected 2 first PCs of sub-pattern sets is approximated by a single line, hence cross-sub-pattern correlation is obvious. Similar kind of cross-sub-pattern correlations are observed between first PCs of sub-pattern sets: **SP2**, **SP3** (Fig. 1(b)) and **SP1**, **SP3** (Fig. 1(c)). Another instance of cross-sub-pattern correlations is seen in musk data (Fig. 2). As shown in Fig. 2 almost entire variance is approximated by a single line showing the presence of cross-sub-pattern correlations and thus a possibility of dimensionality reduction from 2 PCs (features) to 1 feature for this (musk) data set. Thus, we see that cross-sub-pattern correlations are quite common and can be exploited for better summarization of variance.

2.2.2. How to partition a given pattern ?

Partitioning of a pattern should minimize the loss of information due to partitioning, and/or improve the classification. The issue here is the procedure used to divide a pattern. For example, some methods include (i) dividing a pattern by choosing features (equal or different number of features) contiguously in the order of appearance [8], (ii) dividing a pattern by choosing features randomly, (iii) if pattern is an image, we may divide it in different ways such as vertical division or horizontal division or both [14].

2.2.3. Feature order dependency

Consider the selection of p features contiguously, in their order of appearance to form sub-patterns, while dividing the given pattern into k sub-patterns. That is, each instance $X_i = \{x_{i_1}, \ldots, x_{i_p}, \ldots\ldots, x_{i_{(k-1)p+1}}, \ldots, x_{i_{kp}}\}$. However, the classification accuracies may differ with different feature orders since the correlation structures may vary in each sub-pattern set with the feature order. Feature order is a permutation of feature values of a pattern. Optimal performance based upon feature ordering in sub-patterns is an open issue.

2.2.4. Selection of sub-pattern size

A question arises about what should be the sub-pattern size? A simple method is to have all k sub-patterns have the same fixed arbitrary size. If $k=1$, then feature partitioning approaches