Pattern Recognition Letters 31 (2010) 462–468

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

A pre-clustering technique for optimizing subclass discriminant analysis [☆]

Sang-Woon Kim [*]

Department of Computer Science and Engineering, Myongji University, Yongin 449-728, South Korea

ARTICLE INFO

Article history:
Available online 18 July 2009

Keywords:
Dimensionality reduction
Subclass discriminant analysis (SDA)
Linear discriminant analysis (LDA)
Pre-clustering technique

ABSTRACT

Subclass discriminant analysis (SDA) [Zhu, M., Martinez, A.M., 2006. Subclass discriminant analysis. IEEE Trans. Pattern Anal. Machine Intell., 28(8), pp. 1274–1286] is a dimensionality reduction method that has proven successful for different types of class distributions. In SDA, the reduction of dimensionality is not achieved by assuming that each class is represented by a single cluster, but rather by approximating the underlying distribution with a mixture of Gaussians. The advantage of SDA is that since it does not treat the class-conditional distributions as uni-modal ones, the nonlinearly separable problems can be handled as linear ones. The problem with this strategy, however, is that to estimate the number of subclasses needed to represent the distribution of each class, i.e., to find out the best partition, *all* possible solutions should be verified. Therefore, this approach leads to an associated high computational cost. In this paper, we propose a method that optimizes the computational burden of SDA-based classification by simply reducing the number of classes to be examined through choosing a *few* classes of the training set prior to the execution of the SDA. To select the classes to be partitioned, the intra-set distance is employed as a criterion and a k-means clustering is performed to divide them. Our experimental results for an artificial data set of XOR-type samples and three benchmark image databases of Kimia, AT&T, and Yale demonstrate that the processing CPU-time of the SDA optimized with the proposed scheme could be reduced *dramatically* without either sacrificing classification accuracy or increasing computational complexity.

1. Introduction

Even from the infancy of the field of statistical pattern recognition (PR), researchers and practitioners have had to wrestle with the so-called "curse of dimensionality" (Jain et al., 2000). The literature reports numerous strategies that have been used to tackle this problem. The most well known of these is the principal component analysis (PCA) used to compute the basis (eigen) vectors by which the class subspaces are spanned, thus retaining the most significant aspects of the structure in the data (Fukunaga, 1990). While PCA finds components that are efficient for *representation*, the class of linear discriminant analysis (LDA) strategies seeks features that are efficient for *discrimination* (Fukunaga, 1990). LDA methods effectively use the concept of a within-class scatter matrix, S_W, and a between-class scatter matrix, S_B, to maximize a separation criterion, such as $J = tr(S_W^{-1}S_B)$.

Being essentially linear algorithms, neither PCA nor LDA can appropriately handle complex nonlinear data. Consequently, numerous LDA-extensions including two-stage LDA (Belhumeur et al., 1997), direct LDA (Yu and Yang, 2001), regularized LDA (Dai and Yuen, 2003), weighted LDA (Price and Gee, 2005), null-space LDA (Cevikalp et al., 2005), LDA/QR (Ye and Li, 2005), LDA/GSVD (Howland et al., 2006), Chernoff distance-based LDA (Loog and Duin, 2004; Rueda and Herrera, 2008), two-dimensional LDA (Ming and Yuan, 2005; Zheng et al., 2008), and kernel-based LDA (Yang, 2002; Yang et al., 2005) have been proposed in the literature. Beside these, to discover the nonlinear manifold structure, various techniques including locally linear embedding (LLE) (Roweis and Saul, 2000), local discriminant embedding (LDE; Chen et al., 2000), locally linear discriminant analysis (LLDA; Kim and Kittler, 2005), and mixture discriminant analysis (MDA) and its variants (Fraley and Raftery, 1998; Zohar and Aladjem, 2005; Zohar and Aladjem, 2007) have been proposed.

Recently, to solve the manifold-based problem, Zhu and Martinez (2006) proposed an approach called subclass discriminant analysis (SDA), by which the underlying distribution of each class can be approximated with a mixture of Gaussians (Zhu and Martinez, 2006). The basic idea is to represent the data samples of each class by a set of subclasses, capturing most of the variance in the data. In SDA, the major problem to be addressed is to determine the optimal number of Gaussians per class, i.e., the number of subclasses. To obtain a good estimate for the number of subclasses needed to represent each class pdf, the authors use a

[☆] This work was supported by the National Research Foundation of Korea funded by the Korean Government (NRF-2009-0071283). A preliminary version of this paper was presented at the CIARP2008, the 13th Iberoamerican Congress on Pattern Recognition, Havana, Cuba, in September 9–12, 2008.

[*] Tel.: +82 31 330 6437; fax: +82 31 335 9998.
E-mail address: kimsw@mju.ac.kr

0167-8655/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2009.07.007

cross-validation test on the training set. The problem with this strategy, however, is that to estimate the number of subclasses needed to represent the distribution of each class, i.e., to find out the best partition, all possible solutions should be verified. Therefore, this approach has a high computational cost.

In practice, for a given data set of C classes and H subclass divisions per class, the search area for the best solution is *dramatically* increased, on the order of H^C. Thus, the application of SDA is not allowable for PR in which the number of classes is very large. To overcome this limitation, in this paper, we propose a method that optimizes the computational burden of SDA by simply reducing the number of classes to be examined through choosing a few classes of the training set prior to the execution of the SDA.[1] Rather than divide *all* classes of a training set, only a *few* classes of the set are selected for bisection. To choose the classes to be clustered, the distribution variance of each class can be used as a criterion. To measure the variance of the class, the so-called *intra-set* distance is used. The intra-set distance possesses the capability of measuring the unbiased *variances* of components of the given data set. Thus, it has been used to successfully represent the global distribution structure of data sets. The above method is a way of reducing the computational burden of SDA without sacrificing the performance of classifiers designed on the transformed subspace.

On the other hand, the problem of LDA (and some LDA-extensions) is given by the deficiency of the scatter matrices. For example, in LDA, there are at most $C-1$ eigenvectors corresponding to nonzero eigenvalues since the rank of the scatter matrix S_B is bounded from above by $C-1$. Therefore, LDA can only extract $C-1$ features from the original feature space and sometimes the $C-1$ features obtained are not enough for PR in which the data is not linearly separable. However, in SDA, by dividing the data samples of each class into a set of subclasses, we can define scatter matrices whose ranks are (in general) larger than $C-1$ (Zhu and Martinez, 2006).

The main contribution of this paper is to demonstrate that SDA-based classification can be optimized by employing a pre-clustering technique. This has been done by performing the clustering step prior to the SDA process and by demonstrating its strength in terms of the CPU processing time and classification accuracy.

The paper is organized as follows: In Section 2, after providing a brief introduction to the overview of SDA, we present an analysis on the computational complexity of SDA. In Section 3, we present an algorithm that reduces the processing CPU-time of SDA without sacrificing the classification accuracy by utilizing a pre-clustering technique. In Section 4, we present the experimental results of an artificial XOR-type data set and three real-life benchmark databases. In Section 5, we present our concluding remarks.

2. Subclass discriminant analysis (SDA)

2.1. The SDA algorithm

A well-known technique that can solve the problem of *discriminant analysis* is the maximization of the Fisher criterion (the generalized Rayleigh quotient), $J(V)=\frac{|V^TS_BV|}{|V^TS_WV|}$, where V denotes a transformation matrix employed to obtain the subspaces desired and S_B and S_W are a between-class scatter matrix and a within-class scatter matrix, respectively. For example, in LDA, they are defined as

$$S_B=\frac{1}{n}\sum_{i=1}^{C}n_i(\mu_i-\mu)(\mu_i-\mu)^T, \tag{1}$$

$$S_W=\frac{1}{n}\sum_{i=1}^{C}\sum_{j=1}^{n_i}(x_{ij}-\mu_i)(x_{ij}-\mu_i)^T, \tag{2}$$

where C is the number of classes, μ_i the sample mean of class i, μ the global mean, x_{ij} the jth sample of class i, n_i the number of samples in that class, and $n=\sum_{i=1}^{C}n_i$. Here, the LDA algorithm assumes the samples of each class are generated from underlying normal distributions of common covariance matrix but different means. The goal of SDA, however, is to approximate the underlying distribution of each class as a mixture of Gaussians. Once the data distribution of each class has been approximated using a mixture of Gaussians, to find those discriminant vectors that best classify the data, we can use the eigenvalue decomposition equation of $\Sigma_X^{-1}\Sigma_BV=V\Lambda_X$, where Σ_B is a between-subclass scatter matrix, Σ_X is the covariance matrix of the data, V is the transformation matrix whose columns correspond to the discriminant vectors, and Λ_X is a diagonal matrix of the corresponding eigenvalues. Thus, the problem to be addressed is to find that division of the classes into a set of subclasses so that the classification in the reduced space is maximized.

In SDA, to obtain the subspaces for the data, the between-subclass scatter matrix, Σ_B, is defined as

$$\Sigma_B=\sum_{i=1}^{C-1}\sum_{j=1}^{H_i}\sum_{k=i+1}^{C}\sum_{l=1}^{H_k}p_{ij}p_{kl}\left(\mu_{ij}-\mu_{kl}\right)\left(\mu_{ij}-\mu_{kl}\right)^T, \tag{3}$$

where H_i is the number of subclass divisions in class i, and p_{ij} and μ_{ij} are the prior probability and mean of the jth subclass of class i, respectively. Then p_{ij} could be estimated with $\frac{n_{ij}}{n}$, where n_{ij} is the number of samples in the jth subclass of class i.

A formalized SDA algorithm (Zhu and Martinez, 2006) is summarized in the following:

```
1. Initialization: $R_H=0,\forall H$.
2. Perform the following steps from $i=1$ to $n$ by incrementing $i$ per every iteration.
   (a) Generate the training set $X_i=\{x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n\}$ using the NN clustering.
   (b) Compute $\Sigma_X(=S_W)$ of (2) with $X_i$.
   (c) Perform the following steps from $H=C$ to $tC$ by incrementing $H$ per every iteration. Here, $t$ is an experimental constant to guarantee that the minimum number of samples per subclass is sufficiently large.
       (i) Compute $\Sigma_B$ using (3).
       (ii) Perform the eigenvalue decomposition $\Sigma_X^{-1}\Sigma_BV=V\Lambda_X$.
       (iii) Project the data set $X_i$ onto the subspaces of $V_q$, i.e., $Y_i=V_q^TX_i$.
       (iv) If the sample $\boldsymbol{z}_i(=V_q^Tx_i)$ is classified correctly, then $R_H=R_H+1$.
3. Achieve the optimal value of $H$ with $H_o=\arg\max_H\{R_H\}$.
4. After calculating $\Sigma_X$ and $\Sigma_B$ using $X$ and $H_o$, obtain the final projection matrix $V_q^*$ given by the first $q$ columns of $V$.
```

In the above algorithm, the optimal Σ_B^* can be computed with H_o. The resulting projection matrix V_q^* is $d\times q$ matrix whose columns are the eigenvectors associated with the q largest eigenvalues when $H=H_o$, while the dimensionality of the sample vectors is d. Thus, the dimensionality of the projected vectors is $q(\ll d)$. Notice also that the sample to be tested is projected onto the q-dimensional space represented by the projection matrix.

On the other hand, in Step 2(a), the *NN* clustering is performed as follows: First, after constructing an $n_i\times n_i$ Euclidean distance

[1] This strategy has been applied to various applications. An example can be found in (Kim and Duin, 2007).