Journal of Computer Science 2 (9): 754-757, 2006
ISSN 1549-3636

Rotational Linear Discriminant Analysis Using Bayes Rule for Dimensionality Reduction

Alok Sharma, Kuldip K. Paliwal
Signal Processing Lab, Brisbane, Australia

Abstract: Linear discriminant analysis (LDA) finds an orientation that projects high dimensional feature vectors to reduced dimensional feature space in such a way that the overlapping between the classes in this feature space is minimum. This overlapping is usually finite and produces finite classification error which is further minimized by rotational LDA technique. This rotational LDA technique rotates the classes individually in the original feature space in a manner that enables further reduction of error. In this paper we present an extension of the rotational LDA technique by utilizing Bayes decision theory for class separation which improves the classification performance even further.

Key words: Rotational LDA, classification error, Bayes decision theory

INTRODUCTION

Handling of feature vector is quite unmanageable when its dimensionality is very large. It then becomes important to transform these high dimensional feature vectors to reduced feature space for the ease of management. One popular technique for this purpose is linear discriminant analysis (LDA). The LDA technique uses a linear transform to project the feature vectors to a subspace in such a way that the overlapping between different classes (state of the nature) is minimum. However, there may be some finite overlapping which result in finite amount of classification error. This classification error is inevitable for the LDA technique.

Recently rotational LDA technique[1] is presented which minimizes this limitation of LDA technique. In order to minimize the overlaps, it utilizes two transforms: rotational transform θ and orientation W. The rotational transform θ rotates the original feature space in such a way that thereafter the utilization of orientation W produces a reduced feature space which is most discriminative for different classes. The computation of θ is an iterative process which requires some components to be evaluated including the regions in the subspace belonging to classes. The boundaries of these regions are computed using minimum distance classification method. Therefore the boundaries of regions are dependent on the type of the classifier used. Thus the choice of classifier becomes crucial for separating regions in the reduced feature space which influences the overall classification performance. In this paper we have utilized Bayes decision theorem with Gaussian density function for this purpose. The utilization of Bayesian rule seems to improve the performance in terms of getting lesser classification error which is empirically demonstrated. Also an adaptive approach is adopted to compute within-class scatter matrix S_W for the computation of orientation W.

NOTATIONS AND DESCRIPTIONS

In the remaining discussions $\mathcal{X}$ denotes the d-dimensional set of n training samples (feature vectors) in a c-class problem, $\Omega = \{\omega_i : i = 1, 2, ..., c\}$ be the finite set of c states of nature or class labels where ω_i denotes the i^{th} class label. The set $\mathcal{X}$ can be subdivided into c subsets $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_c$ where each subset $\mathcal{X}_i$ belongs to ω_i and consists of n_i number of samples such that:

$$n = \sum_{i=1}^{c} n_i$$

The samples or patterns of set $\mathcal{X}$ can be written as:

$\mathcal{X} = \{x_1, x_2, \cdots, x_n\}$ where $x_j \in R^d$ (d-dimensional hyperplane)

$\mathcal{X}_i \subset \mathcal{X}$ and $\mathcal{X}_1 \cup \mathcal{X}_2 \cup \ldots \cup \mathcal{X}_c = \mathcal{X}$

Let $\mathcal{Y}_j$ be h-dimensional transformed samples from $\mathcal{X}_j \in \omega_j$ using rotational LDA technique where $h < d$, then the samples of reduced dimensional set or transformed sample set $\mathcal{Y}$ can be depicted as:

$\mathcal{Y} = \{y_1, y_2, \cdots, y_n\}$ where $y_j \in R^h$ (h-dimensional hyperplane)

$\mathcal{Y}_1 \cup \mathcal{Y}_2 \cup ... \cup \mathcal{Y}_c = \mathcal{Y}$ and $\mathcal{Y}_1 \cup \mathcal{Y}_2 \cup ... \cup \mathcal{Y}_c = \mathcal{Y}$ where $\mathcal{Y}_j$ is derived from $\mathcal{X}_j$

For convenience, the notations used in the rest of the paper are elaborated as follows:

S_W within-class scatter matrix

S_B between class scatter matrix

W $d \times h$ transformation matrix (orientation)

θ $d \times d$ transformation matrix (rotational)

μ_{x_j} center of $\mathcal{X}_j \in \omega_j$

μ_{y_j} center of $\mathcal{Y}_j \in \omega_j$

Corresponding Author: Alok Sharma, Kuldip K. Paliwal, Signal Processing Lab, Brisbane, Australia

$\mathcal{R}_j$ j^{th} region in the reduced dimensional space (R^h) which belongs to ω_j

Σ_{y_j} covariance of $\mathcal{Y}_j$

A review: Rotational linear discriminant analysis: Rotational linear discriminant analysis (LDA)[1] rotates the individual classes in the original *d*-dimensional feature space such that the overlapping of samples of classes in the reduced *h*-dimensional reduced feature space is minimum (where $h < d$). The technique finds transformation W and $\boldsymbol{\theta}$ of sizes $d \times h$ and $d \times d$ respectively. The transformation or orientation W is a set of *h*, *d*-dimensional column vectors which represent the most discriminative directions in the reduced feature space. The orientation W is obtained by the eigenvalue decomposition of scatter matrices[2]

$$S_W^{-1} S_B \, w_i = \lambda_i \, w_i \quad (1)$$

where w_i is the i^{th} column vector of W. The within-class scatter matrix S_W and between class scatter matrix S_b can be evaluated as:

$$S_b = \sum_{i=1}^{c} n_i (\mu_i - \mu)(\mu_i - \mu)^T \quad (2)$$

$$S_W = \sum_{i=1}^{c} \sum_{x \in \mathcal{X}_i} (x - \mu_i)(x - \mu_i)^T \quad (3)$$

The rotational transform θ consists of *d*-dimensional orthonormal vectors i.e. $\theta^T\theta = I_{d\times d}$. The rotational transform θ enables rotation of samples of classes in the original feature space prior to the application of orientation W.

The rotational LDA technique, transforms a feature vector $x \in R^d$ to $y \in R^h$ by using the relation

$$y = W^T[\theta^T(x - \mu_{x_j}) + \mu_{x_j}] \quad (4)$$

It can be observed from equation 4 that when there is no rotation (i.e. $\theta = I_{d\times d}$) then it turns to be the basic LDA method. The optimum value of θ is computed by minimizing the overlapping of samples in the reduced dimensional plane. The solution for θ can be obtained from the following iterative procedure[1]:

$$\theta \propto \sum_{j=1}^{c} \frac{n_j^2}{|\Sigma_{y_j}|} \underset{x(y \in \mathcal{R}_j)}{E} [F(x, \theta, W, \mu_{x_j}, \Sigma_{y_j})] \quad (5)$$

$$\theta \leftarrow \theta(\theta^T\theta)^{-1/2} \quad (6)$$

where

$F(x, \theta, W, \mu_{x_j}, \Sigma_{y_j}) = \exp(-\frac{1}{2}u)[(x - \mu_{x_j})(x - \mu_{x_j})^T \theta W(\Sigma_{y_j}^{-1} + \Sigma_{y_j}^{-1T}) W^T]$ and $E[\bullet]$ is the expectation of $F(\bullet)$ with respect to x. The correspondence relation $x(y \in \mathcal{R}_j)$ in equation 5 depicts that only those vectors of $x \in \mathcal{X}_j$ are taken which correspond to $y \in \mathcal{R}_j \subset \mathcal{Y}_j$.

The inverse of $\theta^T\theta$ in equation 6 is computed using eigenvalue decomposition. There are iterative methods for orthonormalization that avoid the matrix inverse and eigendecomposition. In that case the rotation matrix θ can be orthonormalized by using symmetric orthonormalization procedure starting from a non-orthogonal matrix and continuing the iterative process until $\theta^T\theta \approx I_{d\times d}$ [3].

The region $\mathcal{R}_j$ for feature vector y is investigated by utilizing the minimum distance classification method on the training feature vector $x \in \mathcal{X}$ in the following manner:

Step 1: Transform feature vector x using orientation W as

$$y = W^T x$$

Step 2: Find distance δ_j using (for example) Euclidean norm

$$\delta_j = \| y - \mu_{y_j} \| \text{ for } j = 1 \ldots c$$

Step 3: Associate feature vector y to the closest region

$$k = \arg\min_{j=1}^{c} \delta_j$$

$$y \in \mathcal{R}_k \in \omega_k$$

Once the region $\mathcal{R}_j$ is defined for y, the overlapping error (Perror) of samples between classes can be obtained as

$$\text{Perror} = 1 - \frac{\sum_{j=1}^{c} \text{number of samples belongs to } \mathcal{R}_j \text{ given } \omega_j}{\text{total number of samples in } \mathcal{X}} = 1 - \frac{\sum_{j=1}^{c} (n_j \mid \mathcal{R}_j, \omega_j)}{n} \quad (7)$$

Separation of reduced feature space into *c* regions using bayes decision theory: The Bayes decision theorem has been incorporated to separate the reduced feature space into *c* disjoint regions ($\mathcal{R}_j$). Bayesian rule produces improved performance for rotational LDA technique in terms of getting lesser classification error when compared with the rotational LDA technique while using minimum distance classification method. The Bayes rule can be given as

$$P(\omega_j \mid y) = \frac{p(y \mid \omega_j) P(\omega_j)}{p(y)} \quad (8)$$

where $p(y \mid \omega_j)$ is probability density function, $P(\omega_j)$ is *a priori* probability, $p(y)$ is probability of occurrence and $P(\omega_j \mid y)$ is *a posteriori* probability. It can be observed from equation 8 that probability of occurrence is independent of the state of nature ω_j and therefore when investigating the membership of feature vector y (corresponding $x \in \mathcal{X}$), $p(y)$ can be discarded. Thus the decision rule is given as

$$k = \arg\max_{j=1}^{c} p(\mathrm{y} \mid \omega_j) P(\omega_j)$$

The Gaussian normal density is taken for $p(\mathrm{y} \mid \omega_j)$. This yields the following algorithm (table 1) for investigating the membership of feature vector $\mathrm{x} \in \mathcal{X}$:

Table 1: An approach to find the membership of feature vector using Bayes rule

Step 1: Transform feature vector x using orientation W as

$\mathrm{y} = \mathrm{W}^{\mathrm{T}}\mathrm{x}$

Step 2: Find the probability

$p_j = \log p(y \mid \omega_j) P(\omega_j) =$

$-\frac{1}{2}(\mathrm{y}-\mu_{\mathrm{y}_j})^{\mathrm{T}} \Sigma_{\mathrm{y}_j}^{-1} (\mathrm{y}-\mu_{\mathrm{y}_j}) - \frac{1}{2}\log|\Sigma_{\mathrm{y}_j}| + \log P(\omega_j)$

for $j = 1 \ldots c$

Note the log function can be used without affecting the decision.

Step 3: Associate feature vector y to the closest region

$k = \arg\max_{j=1}^{c} p_j$

$\mathrm{y} \in \mathcal{R}_k \in \omega_k$

Adaptive within-class scatter matrix for iterative rotational LDA process: The rotation LDA technique computes rotational transform θ iteratively until the probability of error (Perror) is minimized. The computation of θ requires to compute orientation W (equation 5) in each of its iterative step. This means that the evaluation of within-class scatter matrix S_W for every single change in the value of θ. On the other hand, the between class scatter matrix S_b does not require to be evaluated iteratively since it depends on the class centers and total mean vector[2] which remains invariant during the rotation process. It is suggested in Ref.[1] to adaptively update S_W that could be economical in computation which was, however, not discussed in detail. The adaptive method for S_W is given as follows[1]:

$$S_W \leftarrow \hat{\theta}^{\mathrm{T}} S_W \hat{\theta} \qquad (9)$$

where $\hat{\theta}$ is rotation occurred at any arbitrary iteration. The computation complexity of S_W (in equation 9) for any iteration is estimated to be $O(d^3)$. On the other hand, using standard procedure (equation 3) it is estimated to be $O(d^2 n)$. Therefore adaptive method (equation 9) is economical only when $n > d$. If $n < d$ then matrix S_W will become singular and it will not be possible to find orientation W using eigenvalue decomposition method (equation 1). In this case some intermediate techniques like PCA prior to LDA[4-7] can be used which reduces the dimensionality so that the matrix S_W becomes full rank (non-singular) and orientation W can be found using equation 1. This would, however, sacrifice some classification performance. There are some direct methods also available[8,9] which do not require any intermediate techniques and are able to compute the orientation directly.

Extension of rotational LDA technique: This section describes the extended version of the rotational LDA technique. In this extension the regions are separated into c distinct classes in the reduced feature space by Bayes decision theorem using Gaussian normal density function. Also, the within-class scatter matrix is updated adaptively which turns to be economical when $n > d$. The modified algorithm is depicted in Table 2.

Table 2: Rotational LDA algorithm using Bayesian rule for estimating orientation W and rotation θ

1. Find the mean of each class $\mu_{\mathrm{x}_j} \in \mathrm{R}^d$ for $j = 1 \ldots c$.
2. Initialize $\theta \leftarrow \mathrm{I}_{d \times d}$, $\hat{\theta} \leftarrow \mathrm{I}_{d \times d}$, $\mathrm{P}_0 = \mathrm{Perror} \leftarrow 100\%$ and set counter $m \leftarrow 0$.
3. Compute S_b and S_W using equations 2 and 3 respectively.
4. while (true)
5. Increment counter $m \leftarrow m + 1$.
6. Update S_W

   $S_W \leftarrow \hat{\theta}^{\mathrm{T}} S_W \hat{\theta}$
7. Find $\mathrm{W} = \{\mathrm{w}_i : i = 1 \ldots h\}$ using equation 1.
8. Compute transformed samples $\mathrm{y} \in \mathrm{R}^h$ and $\mu_{\mathrm{y}_j} \in \mathrm{R}^h$ (where $j = 1 \ldots c$).
9. Perform classification (to find $\mathcal{R}_j$) from table 1 and compute P_m (new Perror) using equation 7.
10. Stop the iterative procedure if P_m is not decreasing

    if ($\mathrm{P}_m > \mathrm{P}_{m-1}$)

    break

    end
11. Store orientation matrix $\hat{\mathrm{W}} \leftarrow \mathrm{W}$ and center of each class $\mu_{\mathrm{y}_j} \in \mathrm{R}^h$.
12. Compute covariance Σ_{y_j} (where $j = 1 \ldots c$).
13. Update rotation matrix

    $\theta \leftarrow \theta\hat{\theta}$

    $\hat{\theta} \leftarrow \sum_{j=1}^{c} \frac{n_j^2}{|\Sigma_{\mathrm{y}_j}|} E_{\mathrm{x}(\mathrm{y} \in \mathcal{R}_j)} [\mathrm{F}(\mathrm{x}, \hat{\theta}, \mathrm{W}, \mu_{\mathrm{x}_j}, \Sigma_{\mathrm{y}_j})]$
14. Orthonormalize rotation matrix

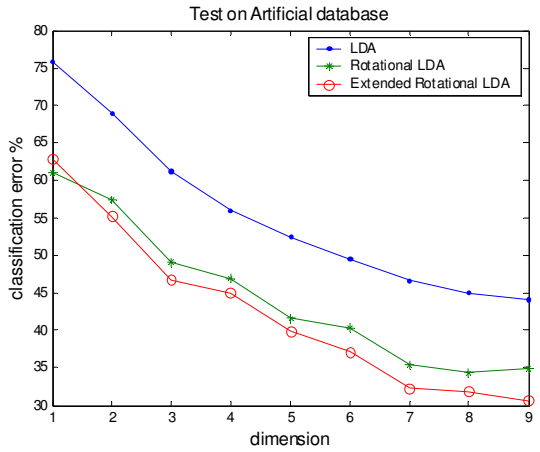
    $\hat{\theta} \leftarrow \hat{\theta}(\hat{\theta}^{\mathrm{T}}\hat{\theta})^{-1/2}$

    $\theta \leftarrow \theta(\theta^{\mathrm{T}}\theta)^{-1/2}$
15. Update rotation matrix $\mathcal{X}_j \leftarrow \hat{\theta}^{\mathrm{T}}(\mathcal{X}_j - \mu_{\mathrm{y}_j}) + \mu_{\mathrm{y}_j}$ for $j = 1 \ldots c$
16. end

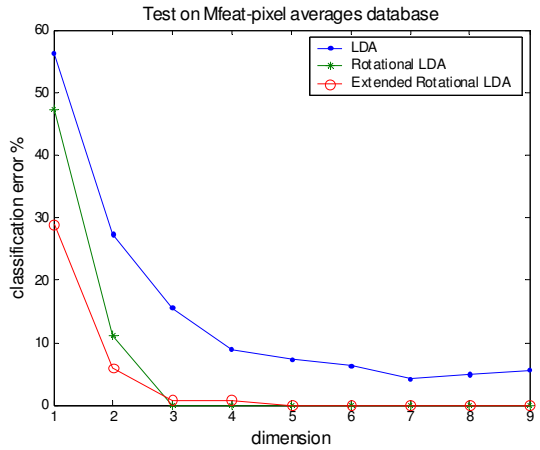
EXPERIMENTATION

This section illustrates the performance of the extended rotational LDA technique in comparison with basic LDA and rotational LDA techniques. To verify the proposed extension we performed two experiments, first on artificial data and second on real data.

For the artificial case, a set of 10 classes is generated from a 30-dimensional normal distribution



(a)



(b)

Fig. 1: Classification error as a function of dimension on (a) artificial dataset and on (b) Mfeat-pixel averages dataset

with different covariance matrices Σ_{x_j} and known class means, where Σ_{x_j} is taken randomly (e.g. $\Sigma_{x_j} = rand(30,30) * j$). The number of feature vectors are equal for all the classes, thus the *a priori* probabilities are equal. The reduced subspace of dimensions 1 to 9 is obtained, for which classification error is computed. The classification error as a function of dimension for all the three methods on artificial data is depicted on Fig. 1a. It can be observed from Fig. 1a that extended rotational LDA is producing better classification error when compared with the rotational LDA technique and with the basic LDA technique.

For the real dataset, multiple features (Mfeat) dataset for pixel averages[10,11] are used. This is a 10-class corpus with 240 dimensions. A sum of 1500 feature vectors is used for training the classifier and a separate set of 500 feature vectors are used for testing. Three methods are used again to verify the performance in terms of classification error. Figure 1b illustrates the resulting classification errors as a function of dimensions. It can be seen from Fig. 1b that extended rotational LDA is producing better results than rotational LDA technique especially at low dimensions (1 to 3) and approximately same thereafter. It is also evident that the extended rotational LDA technique is producing better results than the basic LDA technique.

CONCLUSION

We have presented an extension of the rotational LDA technique which is producing lesser classification error when compared with the rotational LDA and the basic LDA techniques on artificial and on real datasets. The within-class scatter matrix is computed in an adaptive fashion which is economical when the number of training samples is higher then the feature dimensions.

REFERENCES

1. Sharma, A. and K.K. Paliwal, 2006. Rotational Linear Discriminant Analysis Technique for Dimensionality Reduction (under review).
2. Duda, R.O. and P.E. Hart, 1973. Pattern Classification and Scene Analysis. John Wiley and Sons, New York.
3. Hyvärinen, A., 1999. Fast and robust fixed-point algorithms for independent component analysis. IEEE Trans. on Neural Networks, 10: 626-634.
4. Swets, D.L. and J. Weng, 1996. Using discriminative eigenfeatures for image retrieval. IEEE Trans. Pattern Analysis and Machine Intelligence, 18: 831-836.
5. Belhumeur, P.N., J.P. Hespanha and D.J. Kriegman, 1997. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Trans. Pattern Analysis and Machine Intelligence, 19: 711-720.
6. Zhao, W., R. Chellappa and N. Nandhakumar, 1998. Empirical performance analysis of linear discriminant classifiers. Proceedings. IEEE Conf. on Computer Vision and Pattern Recognition, pp: 164-169.
7. Sharma, A., K.K. Paliwal and G.C. Onwubolu, 2006. Class-dependent PCA, MDC and LDA: A Combined Classifier for Pattern Classification. Pattern Recognition, 39: 1215-1229.
8. Chen, L.-F., H.-Y.M. Liao, M.-T.Ko, J.-C. Lin and G.-J. Yu, 2000. A new LDA-based face recognition system which can solve the small sample size problem. Pattern Recognition, 33: 1713-1726.
9. Yu, H. and J. Yang, 2001. A direct LDA algorithm for high-dimensional data-with application to face recognition. Pattern Recognition, 34: 2067-2070.
10. Jain, A.K., R.P.W. Duin and J. Mao, 2000. Statistical pattern recognition: a review. IEEE Trans. on Pattern Anal. Machine Intelligence, 22: 4-37.
11. Blake, C.L. and C.J. Merz, 1998. UCI repository of machine learning databases, http://www.ics.uci.edu/~mlearn, Irvine, CA. University of Calif., Dept. of Information and Comp. Sci.