IMPROVED PSEUDOINVERSE LINEAR DISCRIMINANT ANALYSIS METHOD FOR DIMENSIONALITY REDUCTION

KULDIP K. PALIWAL* and ALOK SHARMA*,†‡§¶

*Signal Processing Laboratory, School of Engineering
Griffith University, QLD-4111, Brisbane, Australia
†University of the South Pacific, Fiji
‡Laboratory of DNA Information Analysis
Human Genome Center, Institute of Medical Science
University of Tokyo, 4-6-1 Shirokanedai
Minato-ku, Tokyo 108-8639, Japan
§aloks@ims.u-tokyo.ac.jp
¶sharma_al@usp.ac.fj

Received 4 November 2010
Accepted 22 September 2011
Published 11 May 2012

Pseudoinverse linear discriminant analysis (PLDA) is a classical method for solving small sample size problem. However, its performance is limited. In this paper, we propose an improved PLDA method which is faster and produces better classification accuracy when experimented on several datasets.

Keywords: Pseudoinverse; linear discriminant analysis; dimensionality reduction; computational complexity.

1. Introduction

Dimensionality reduction is an important aspect of pattern classification. It helps in improving the robustness (or generalization capability) of the pattern classifier and in reducing its computational complexity. The linear discriminant analysis (LDA) method5 is a well-known dimensionality reduction technique studied in the literature. The LDA technique finds an orientation matrix W that transforms high-dimensional feature vectors belonging to different classes to lower dimensional feature vectors such that the projected feature vectors of a class are well separated from the feature vectors of other classes. The orientation W is obtained by maximizing the Fisher’s criterion function $J_1(W) = |W^T S_B W|/|W^T S_W W|$, where $S_B$ is between-class scatter matrix and $S_W$ is within-class scatter matrix. It has been shown in the literature that modified version of Fisher’s criterion $J_2(W) = |W^T S_B W|/|W^T S_T W|$ produces similar results, where $S_T$ is total scatter matrix.6
In the conventional LDA technique, the matrix $S_W$ or matrix (depending upon the criterion taken) needs to be nonsingular. However, in many pattern classification applications these matrices become singular. This problem is known as small sample size (SSS) problem. In order to overcome this problem, several methods have been proposed in the literature. Among these methods, the pseudoinverse LDA (PLDA) method stands as a forerunner and a classical method for solving SSS problem. The PLDA method has been widely studied. It finds the orientation matrix $W$ by computing eigenvalue decomposition (EVD) of $S_W^+S_B$, where $S_W^+$ is the pseudoinverse of $S_W$. However, this has a problem that its computational complexity is $O(d^3)$, which is prohibitively high (when the dimensionality $d$ is very large). Due to this reason, the PLDA method has been cited in the literature in several papers but hardly compared with other techniques for pattern classification. In order to reduce this computational complexity, Liu et al. introduced a fast PLDA method. In their method the orientation $W$ is computed by finding the range space of $S_W$ followed by the range space of $S_B$. The null space of $S_W$ is discarded in this process. Their fast PLDA method has been shown equivalent to the PLDA method. Though the fast PLDA method is computationally faster than the PLDA method, it has a drawback. It discards null space of $S_W$ in computing the orientation matrix $W$, which has been shown to contain useful discriminant information for classification. Considering this drawback we propose the use of modified Fisher’s criterion $J_2(W) = |W^TS_BW|/|W^TS_TW|$ for the pseudoinverse method. Therefore, in the proposed method the orientation $W$ is computed by finding the range space of $S_T$ followed by the range space of $S_B$. In this method the null space of $S_T$ has been discarded. It is known that discarding the null space of $S_T$ does not cause any loss of discriminant information. Thus, this has an advantage over the fast PLDA method that it improves the classification performance. In addition, it is shown to be computationally faster than fast PLDA method.

2. Improved PLDA Method

In order to describe improved PLDA method, we first define some notations. Let $\mathbf{x}$ be a set of $n$ training vectors in a $d$-dimensional feature space, and $\Omega = \{ \omega_i : i = 1, 2, \ldots, c \}$ be the finite set of $c$ class labels, where $\omega_i$ denotes the $i$th class label. The set $\mathbf{x}$ can be subdivided into $c$ subsets $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_c$ (where subset $\mathbf{x}_i$ belongs to $\omega_i$); i.e. $\mathbf{x}_i \subseteq \mathbf{x}$ and $\mathbf{x}_1 \cup \mathbf{x}_2 \cup \cdots \cup \mathbf{x}_c = \mathbf{x}$. Let $n_i$ be the number of samples in class $\omega_i$ such that:

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

All these methods except the method by Zhang et al. try to maximize Fisher’s criterion or modified Fisher’s criterion either in one stage or in two stages. In the Zhang’s method, the difference between $S_W^tS_B$ and $WDW^t$ is minimized, where $W$ is an orthogonal matrix and $D$ is a diagonal matrix. This method also deals with the case when $N > d$ (where $N$ is the number of samples and $d$ is the dimensionality).
The samples or vectors of set \( \mathcal{X} \) can be written as:
\[
\mathcal{X} = \{x_1, x_2, \ldots, x_n\}, \quad \text{where } x_j \in \mathbb{R}^d.
\]
Let \( \mu_j \) be the centroid of \( \mathcal{X}_j \) and \( \mu \) be the centroid of \( \mathcal{X} \), then the between-class scatter matrix \( S_B \) is given by
\[
S_B = \sum_{j=1}^{c} n_j (\mu_j - \mu)(\mu_j - \mu)^T.
\]
The within-class scatter matrix \( S_W \) is defined as
\[
S_W = \sum_{j=1}^{c} S_j,
\]
where \( S_j = \sum_{x \in \mathcal{X}_j} (x - \mu_j)(x - \mu_j)^T \).

The total-class scatter matrix \( S_T \) is defined as
\[
S_T = \sum_{j=1}^{n} (x_j - \mu)(x_j - \mu)^T.
\]
The matrix \( S_T \) can also be formed as \( S_T = AA^T \), where \( A \in \mathbb{R}^{d \times n} \) is defined as
\[
A = [(x_1 - \mu), (x_2 - \mu), \ldots, (x_n - \mu)].
\]
In a similar way, \( S_B \) can be formed as \( S_B = BB^T \), where rectangular matrix \( B \in \mathbb{R}^{d \times c} \) can be defined as
\[
B = [\sqrt{n_1}(\mu_1 - \mu), \sqrt{n_2}(\mu_2 - \mu), \ldots, \sqrt{n_c}(\mu_c - \mu)].
\]

Let the ranks of matrices \( S_T, S_B, \) and \( S_W \) be \( t, b \) and \( w \), respectively.

The orientation matrix \( W \) can be obtained by first finding the range space of \( S_T \) followed by the range space of \( S_B \), i.e. if EVD of \( S_T \) is
\[
S_T = U_1 \Lambda U_1^T,
\]
where \( U_1 \in \mathbb{R}^{d \times t} \) corresponds to the range space of \( S_T \) and \( \Lambda \in \mathbb{R}^{t \times t} \) is a diagonal matrix, then \( \hat{S}_T = U_1^T S_T U_1 \) and \( \hat{S}_B = U_1^T S_B U_1 \), the orientation matrix \( W \) can be found by finding EVD of \( \hat{S}_T^T \hat{S}_B \).

In order to find the range of \( S_T \), we can compute EVD of \( A^T A \in \mathbb{R}^{n \times n} \) instead of \( S_T = AA^T \in \mathbb{R}^{d \times d} \), this will significantly reduce the computational complexity.\(^6\)

If the eigenvectors and eigenvalues of \( A^T A \in \mathbb{R}^{n \times n} \) are \( E \in \mathbb{R}^{n \times n} \) and \( D \in \mathbb{R}^{n \times n} \), respectively, then
\[
A^T A = E D E^T
\]
\[
= \begin{bmatrix} E_1, E_2 \end{bmatrix} \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} \begin{bmatrix} E_1^T \\ E_2^T \end{bmatrix},
\]
where \( E_1 \in \mathbb{R}^{n \times t}, E_2 \in \mathbb{R}^{n \times (n-t)} \) and \( D_1 \in \mathbb{R}^{t \times t} \)
\[
= E_1 D_1 E_1^T
\]
\( (2) \)
and orthonormal eigenvectors $U_1$ defining the range space of $S_T$ can be given as

$$U_1 = AE_1D_1^{-1/2}.$$  

Since discarding the null space of $S_T$ does not cause any loss of discriminant information, we can use $U_1 \in \mathbb{R}^{d \times t}$ to transform the original $d$-dimensional space to a lower $t$-dimensional space. The matrices $A$ and $B$ can be written in the lower dimensional space as follows:

$$\hat{A} = U_1^T A \in \mathbb{R}^{t \times n}$$

$$= D_1^{1/2}E_1^TA$$

$$= D_1^{1/2}E_1^TE_1D_1^{1/2} = D_1^A \quad (\text{from Eq. (2)})$$

$$= D_1^{1/2}E_1^T.$$  

(3)

The matrix $\hat{B}$ can be economically constructed from $\hat{A}$. In order to do this, we first write the transformed matrix $\hat{A}$ as $\hat{A} = [v_1, v_2, \ldots, v_n]$ and then compute $\hat{B}$ as

$$\hat{B} = \left[ \frac{1}{\sqrt{n_1}} \sum_{j=1}^{n_1} v_j, \frac{1}{\sqrt{n_2}} \sum_{j=n_1+1}^{n_1+n_2} v_j, \ldots, \frac{1}{\sqrt{n_c}} \sum_{j=n_1+n_2+\ldots+n_{c-1}+1}^{n} v_j \right].$$  

(4)

This will give transformed between-class scatter $\hat{S}_B = \hat{BB}^T$. From Eq. (3), the transformed total-scatter matrix $\hat{S}_T = \hat{AA}^T = D_1^{1/2}E_1^TE_1D_1^{1/2} = D_1^A$, this will give $\hat{S}_T = \hat{S}_B = D_1^{-1}\hat{B}\hat{B}^T$. The EVD of $\hat{S}_T^+\hat{S}_B$ will give eigenvectors $W \in \mathbb{R}^{t \times h}$ (where $h$ is less than or equal to the rank of $\hat{S}_T^+\hat{S}_B$, in other words, $1 \leq h \leq c-1$) corresponding to its leading eigenvalues. The orientation matrix $W$ can be obtained as follows:

$$W = U_1W = AE_1D_1^{-1/2}W.$$  

The implementation of the improved PLDA method is summarized in Table 1.

### 3. Computational Complexity and Storage Requirements

In this section, the computational complexity and storage requirements of the proposed improved PLDA method are discussed. We also compare its computational complexity and storage requirements with PLDA and fast PLDA methods. The

<table>
<thead>
<tr>
<th>Table 1. Improved PLDA method.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Construct matrix $A$ from Eq. (1).</td>
</tr>
<tr>
<td>2. Compute eigenvalues $E_1 \in \mathbb{R}^{n \times t}$ and eigenvectors $D_1 \in \mathbb{R}^{t \times t}$ of $A^T A \in \mathbb{R}^{n \times n}$.</td>
</tr>
<tr>
<td>3. Compute transformed matrix $\hat{B}$ (from Eq. (4)).</td>
</tr>
<tr>
<td>4. Compute the EVD of $D_1^A \hat{B} \hat{B}^T$ to get $W \in \mathbb{R}^{t \times h}$ (where $1 \leq h \leq c-1$).</td>
</tr>
<tr>
<td>5. Compute $W = D_1^{-1/2}\hat{W}$, then $W \leftarrow E_1W$ and then $W \leftarrow AW$.</td>
</tr>
</tbody>
</table>
estimated computational complexity of the improved PLDA method is listed in Table 2.

Since the dimensionality $d$ in a SSS problem is very large compared to the number of training samples $n$ ($d \gg n$), the computational complexity of the improved PLDA method boils down to $dn^2 + dnc + 2dn$ flops.

In the PLDA method, the computation of EVD of $S_W = S_B$ is required. This has the computational complexity of $O(d^3)$. The fast PLDA method requires approximately $3dn^2 + 2dnc + 3dn$ flops. When the dimensionality is very large $d \gg n$ then the proposed method is approximately three times faster than the fast PLDA method. The storage requirements of all the methods are same. In all the methods, the orientation matrix $W \in \mathbb{R}^{d \times h}$ computed during training session is required to be stored for the testing session which requires approximately $dh$ storage.

### 4. Datasets and Experimentation

The following types of datasets are utilized for the experimentation: DNA microarray gene expression data, face recognition data and text classification data. We have also used randomly generated data to investigate the effect of dimensionality $d$ on the computation time. Five DNA microarray gene expression datasets are utilized. We use the splitting of the data into the training and test samples as provided by the distributors.\(^b\) For face recognition, AR database\(^{12}\) is utilized for the experimentation. A subset of AR database is used here with 1400 face images from 100 persons (14 images per person). Training set contains seven images per person and

---

the remaining seven images per person are used for testing. The dimensionality $d$ is 19,800. We use a subset of Dexter dataset\textsuperscript{3} for text classification in a bag-of-word representation. This dataset has sparse continuous input variables. The description of all the datasets is given in Table 3.

The fast PLDA method and the improved PLDA method have been experimented on all the above datasets. In addition, Fisherface LDA method\textsuperscript{17} and null space-based method\textsuperscript{19} have been used for comparison purpose. The nearest neighbor classifier has been used to classify test feature vector. The classification accuracy and CPU time of these methods are depicted in Table 4. It can be observed from Table 4 that improved PLDA method is outperforming fast PLDA method in terms of classification accuracy and CPU time. Furthermore, the improved PLDA method is computationally efficient than Fisherface LDA and null space-based methods. It can also be observed (in terms of classification accuracy) that improved PLDA method is outperforming Fisherface LDA method and proving as good as results with the null space-based method.

We have also generated random data and increased its dimensionality from 10,000 to 100,000, to measure the CPU time of the improved PLDA and fast PLDA methods. The CPU time curve as a function of data dimensionality is shown in Fig. 1. It can be seen from the figure that as the dimensionality of data increases the improved PLDA method performs faster than the fast PLDA method.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Class</th>
<th>Dimension</th>
<th>Number of training samples</th>
<th>Number of testing samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acute Leukemia\textsuperscript{7}</td>
<td>2</td>
<td>7129</td>
<td>38</td>
<td>34</td>
</tr>
<tr>
<td>ALL Subtype\textsuperscript{20}</td>
<td>7</td>
<td>12558</td>
<td>215</td>
<td>112</td>
</tr>
<tr>
<td>MLL\textsuperscript{1}</td>
<td>3</td>
<td>12582</td>
<td>57</td>
<td>15</td>
</tr>
<tr>
<td>GCM\textsuperscript{14}</td>
<td>14</td>
<td>16063</td>
<td>144</td>
<td>54</td>
</tr>
<tr>
<td>Lung Cancer\textsuperscript{5}</td>
<td>2</td>
<td>12533</td>
<td>32</td>
<td>149</td>
</tr>
<tr>
<td>Face AR\textsuperscript{12}</td>
<td>100</td>
<td>19800</td>
<td>700</td>
<td>700</td>
</tr>
<tr>
<td>Dexter\textsuperscript{3}</td>
<td>2</td>
<td>20000</td>
<td>300</td>
<td>300</td>
</tr>
</tbody>
</table>

Table 3. Datasets used in the experimentation.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Fast PLDA method</th>
<th>Improved PLDA method</th>
<th>Fisherface LDA method</th>
<th>Null space-based method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Classn. accuracy</td>
<td>CPU time</td>
<td>Classn. accuracy</td>
<td>CPU time</td>
</tr>
<tr>
<td>Acute Leukemia</td>
<td>88.2</td>
<td>0.05</td>
<td>97.1</td>
<td>0.02</td>
</tr>
<tr>
<td>ALL Subtype</td>
<td>59.8</td>
<td>1.00</td>
<td>85.7</td>
<td>0.53</td>
</tr>
<tr>
<td>MLL</td>
<td>80</td>
<td>0.15</td>
<td>100.0</td>
<td>0.06</td>
</tr>
<tr>
<td>GCM</td>
<td>46.3</td>
<td>0.09</td>
<td>70.4</td>
<td>0.29</td>
</tr>
<tr>
<td>Lung Cancer</td>
<td>94.6</td>
<td>0.09</td>
<td>98.0</td>
<td>0.03</td>
</tr>
<tr>
<td>Face AR14</td>
<td>83.1</td>
<td>19.17</td>
<td>88.9</td>
<td>10.47</td>
</tr>
<tr>
<td>Dexter</td>
<td>67.3</td>
<td>2.13</td>
<td>93.7</td>
<td>1.19</td>
</tr>
</tbody>
</table>

Table 4. Classification accuracy (in percentage) and CPU time on datasets.
5. Conclusion

An improved PLDA method has been proposed in this paper. It is outperforming other pseudoinverse methods in terms of computation complexity and classification accuracy when experimented on several datasets.

References


Fig. 1. CPU time as a function of data dimensionality.


Kuldip K. Paliwal received the B.S. degree from Agra University, Agra, India, in 1969, the M.S. degree from Aligarh Muslim University, Aligarh, India, in 1971 and the Ph.D. degree from Bombay University, Bombay, India, in 1978.

He has been carrying out research in the area of speech processing since 1972. He has worked at a number of organizations including Tata Institute of Fundamental Research, Bombay, India, Norwegian Institute of Technology, Trondheim, Norway, University of Keele, U. K., AT&T Bell Laboratories, Murray Hill, New Jersey, U.S.A., AT&T Shannon Laboratories, Florham Park, New Jersey, U.S.A., and Advanced Telecommunication Research Laboratories, Kyoto, Japan. Since July 1993, he has been a professor at Griffith University, Brisbane, Australia, in the School of Microelectronic Engineering. His current research interests include speech recognition, speech coding, speaker recognition, speech enhancement, face recognition, image coding, pattern recognition and artificial neural networks. He has published more than 300 papers in these research areas.

Dr. Paliwal is a Fellow of Acoustical Society of India. He has served the IEEE Signal Processing Society’s Neural Networks Technical Committee as a founding member from 1991 to 1995 and the Speech Processing Technical Committee from 1999 to 2003. He was an Associate Editor of the IEEE Transactions on Speech and Audio Processing during the periods 1994–1997 and 2003–2004. He also served as Associate Editor of the IEEE Signal Processing Letters from 1997 to 2000. He is also in the Editorial Board of the IEEE Signal Processing Magazine. He was the General Co-Chair of the Tenth IEEE Workshop on Neural Networks for Signal Processing (NNSP2000). He has co-edited two books: Speech Coding and Synthesis (published by Elsevier), and Speech and Speaker Recognition: Advanced Topics (published by Kluwer). He has received IEEE Signal Processing Society’s best (senior) paper award in 1995 for his paper on LPC quantization. He is currently serving the Speech Communication journal (published by Elsevier) as its Editor-in-Chief.

Alok Sharma received the B.Tech degree from the University of the South Pacific (USP), Suva, Fiji, in 2000 and the MEng degree, with an academic excellence award, and the Ph.D. degree in the area of pattern recognition from Griffith University, Brisbane, Australia, in 2001 and 2006, respectively. He is currently a research fellow at the University of Tokyo. He is also with the Signal Processing Laboratory, Griffith University and the University of the South Pacific. He participated in various projects carried out in conjunction with Motorola (Sydney), Auslog Pty. Ltd. (Brisbane), CRC Micro Technology (Brisbane), and the French Embassy (Suva). His research interests include pattern recognition, computer security, and human cancer classification. He reviewed several articles from journals like IEEE Trans. NN, IEEE Trans. SMC, Part A: SH, IEEE Journal on STSP, IEEE Trans. KDE, IEEE Trans. EC, Computers and Security, Pattern Recognition, etc. He is a member of IEEE.