IMPROVING CLASSIFICATION PERFORMANCE OF LINEAR FEATURE EXTRACTION ALGORITHMS

Moataz El Ayadi and Konstantinos N. Plataniotis

Multimedia Lab, The Edward S. Rogers Dept. of Electrical and Computer Engineering, University of Toronto, Ontario, Canada M5S 3G4

ABSTRACT

In this work, we propose a new and novel framework for improving the performance of linear feature extraction (LFE) algorithms, characterized by the Bayesian error probability (BEP) in the extracted feature domain. The proposed framework relies on optimizing a tight quadratic approximation to the BEP in the transformed space with respect to the transformation matrix. Applied to many synthetic multi-class Gaussian classification problems, the proposed optimization procedure significantly improves the classification performance when it is initialized by popular LFE matrices such as the Fisher linear discriminant analysis.

Index Terms— Bayesian error probability, linear feature extraction, multivariate Gaussian density.

1. INTRODUCTION

Linear feature extraction (LFE) has been extensively applied in various signal processing applications such as face recognition [1], speech recognition [2], speaker identification [3], micro-array classification, image analysis, and many others. The key advantages of LFE are the simplicity of implementation and the preservation of the shape of the class conditional densities after transformation. Since the introduction of the classical linear discriminant analysis (LDA) by Fisher [4], many extensions have been proposed to account for multiple classes and heteroscedastic classification [5, 6]. However, very few of LFE methods considered the Bayesian error probability (BEP) as a design criterion, though the BEP is the main performance measure of interest in most signal processing applications. In fact, the BEP in the transformed domain is generally a multi-modal function in the transformation matrix. Hence, it is not expected to find an LFE that is superior to all other LFE methods for all classification problems.

Therefore, rather than devising a new LFE method which is not guaranteed to outperform all other LFE methods for all classification problems, the main objective in this paper is to design a new framework for improving the performance of existing LFE methods in terms of the BEP in the transformed domain. We believe that this framework will increase the usability of LFE methods for signal processing applications. The main focus in this paper will be on Gaussian classification problems. The main assumptions are: (1) the number of classes, \( C \), is known before classification, (2) the given classes are disjoint and exhaustive; i.e., a pattern \( x \) belongs to exactly one of the \( C \) classes, and (3) all the class conditional densities and the class prior probabilities are exactly known.

Since the BEP has no closed-form expression, we propose a tight quadratic approximation which can be efficiently optimized using gradient-based techniques.

The paper is divided into five sections. Section 1 is the introduction. The mathematical formulation of the LFE problem is given in section 2. In section 3, the proposed quadratic approximation to the BEP is derived for Gaussian classification problems. Finally, simulation results are shown in section 4 and important conclusions are drawn in section 5.

2. MATHEMATICAL FORMULATION

Consider a signal processing application in which the main task is to distinguish between \( C \) classes; e.g. a face recognition problem with \( C \) different faces. Assume that all patterns are real \( n \)-dimensional vectors, \( x \in \mathbb{R}^n \). In multi-class classification, we are interested in classifying a pattern \( x \) into one of the classes \( \mathcal{H}_1, \ldots, \mathcal{H}_C \) with prior probabilities \( P(\mathcal{H}_1), P(\mathcal{H}_2), \ldots, P(\mathcal{H}_C) \). Patterns of \( \mathcal{H}_i \) are generated according to the Gaussian density \( \mathcal{N}(\mu_i, \Sigma_i) \), i.e.,

\[
    f_x(x|\mathcal{H}_i) = \frac{1}{(2\pi)^{n/2} |\Sigma_i|^{1/2}} e^{-\frac{1}{2} (x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)},
\]

where \( \mu_i \) and \( \Sigma_i \) are the mean vector and the covariance matrix of \( x \) if \( \mathcal{H}_i \) is true, respectively. We assume all \( f_x(x|\mathcal{H}_i) \) to be exactly known before classification as well as \( P(\mathcal{H}_i), i = 1, \ldots, C \). Further, assume that, for each class, all the generated patterns are statistically independent and identically distributed.

When all the class-conditioned densities are known, the following Bayesian decision rule is usually used for classification [4]:

\[
    x \in C_i \text{ if } i = \arg \max_{j=1,\ldots,C} P(\mathcal{H}_j) f_x(x|\mathcal{H}_j),
\]

A common measure of the classification performance is the BEP given by [4]

\[
    P_{eq} = 1 - \int_{x=1,\ldots,C} \max_{i=1,\ldots,C} \left( P(\mathcal{H}_i|x) f_x(x) \right) dx,
\]

where \( P(\mathcal{H}_i|x) \) is the posterior probability of \( \mathcal{H}_i \) given \( x \) and the function \( f_x(x) \) is the unconditional density of \( x \).

We now consider LFE to overcome the estimation and computation problems usually encountered when \( n \) is high.
The motivation for the quadratic approximation is two folds. First, the derived approximation does not involve complex decision boundaries as in the BEP and hence it is simpler to calculate. Second, $Q_y$ can be easily differentiated with respect to the transformation matrix $A$ unlike $P_{xy}$ which involves complex decision boundaries which are also functions of $A$.

In fact, analytical solution of (7) is not feasible. However, an approximate solution can be found by replacing the objective function in (7) by a tight upper bound which can be analytically optimized. Because of the limited size of the manuscript, only the final results are mentioned. The solution to the approximate optimization problem is $\phi_0^* = (7C + 1)/(8C)$ and $\phi_2^* = -1$. The corresponding absolute difference is upper bounded by $(C - 1)/(8C)$; i.e.,

$$|P_{xy} - Q_y(\phi_0^*, \phi_2^*)| \leq \frac{C - 1}{8C}. \quad (8)$$

Substituting $\phi_0^*, \phi_2^*$ in (6) and simplifying, the quadratic approximation is given by

$$Q_y(\phi_0^*, \phi_2^*) = \frac{7C + 1}{8C} - \sum_{i=1}^{C} P(H_i) \int P(H_i|y) f_y(y|x_1) dy. \quad (9)$$

The function $Q_y(\phi_0^*, \phi_2^*)$ has no closed-form expression but can be approximated by any off-the-shelf Monte-Carlo method such as rejection sampling or importance sampling. In this paper, we select ordinary Monte Carlo.

$$Q_y(\phi_0^*, \phi_2^*) \approx \frac{7C + 1}{8C} - \sum_{i=1}^{C} \frac{P(H_i)}{N_i} \sum_{n=1}^{N_i} P(H_i|x_{ni}), \quad (10)$$

where the patterns $\{x_{ni}\}$, $n = 1, \ldots, N_i$ are randomly generated according to the density $f_x(x|H_i)$ and $N_i$ should be sufficiently large. Unlike the BEP, the derivative of $Q_y$ can be efficiently estimated using the Monte-Carlo method via

$$\frac{\partial Q_y(\phi_0^*, \phi_2^*)}{\partial A} \approx - \sum_{i=1}^{C} \frac{P(H_i)}{N_i} \sum_{n=1}^{N_i} F_i(A, x_{ni}). \quad (11)$$

$$F_i(A, x) = P(H_i|Ax) \left( \frac{\partial}{\partial A} \log f_y(Ax|H_i) \right) - \sum_{j=1}^{C} P(H_j|Ax) \left( \frac{\partial}{\partial A} \log f_y(Ax|H_j) \right). \quad (12)$$

---

1The proof of (11) is lengthy and hence omitted from the paper.
Algorithm 1 Estimation of the optimal LFE matrix for Gaussian classification problems.

Require: $C$, $n$, $k$, $P(H_i)$, $\mu_i$, $\Sigma_i$, $i = 1, \ldots, C$.
Output: Optimal transformation matrix $A$.
1: Calculate $M$ in (5).
2: Find $k_{\min} = \text{rank}(M)$.
3: if $k \geq k_{\min}$ then
4:  Find $A$ by performing full rank decomposition to $M = A^T G$.
5: else
6:  Optimize $Q_y$ in (10) w.r.t. $A$. Initialize the optimization algorithm using any other popular LFE algorithm.
7: end if

It is easy to verify that, for the multivariate Gaussian distribution given in (1), the following expression holds:
\[
\frac{\partial}{\partial A} \log f_x(Ax|H_i) = 2 \Sigma_i^{-1} A.
\]
\[
\left[ -\Sigma_i + (x - \mu_i)(x - \mu_i)^T (I - A^T \Sigma_i^{-1} A) \right],
\]  
(13)
where $\Sigma_i = A \Sigma_i A^T$. Thus, when $k < k_{\min}$, we can obtain a suboptimal transformation matrix by optimizing the quadratic approximation in (10) where the gradient is calculated using (11)-(13). In this work, we prefer to optimize $Q_y(\phi_0^*, \phi_2^*)$ using quasi-Newton methods because they have fast convergence and reasonable memory requirements for moderate size problem [10]. Another important issue is the proper initialization for the optimizer. Similar to almost all optimization algorithms, poor initialization may lead to undesired local optimum of $P_{cx}$. In order to alleviate this problem, we suggest initializing the proposed optimization algorithm using any other powerful LFE algorithm such as the Fisher LDA [4] or an approximate decomposition of $M$ in (5) such as the Tubbs method [8]. This way, it is guaranteed that the obtained LFE is not inferior to those initializers. The overall optimization algorithm based on this strategy is depicted in Algorithm 1. In order to reduce the effect of initialization even more, different initializers can be employed and the one resulting in the lowest BEP is selected.

4. SIMULATION RESULTS
We mainly have two sets of simulations. In subsection 4.1, the objective is to study the quality of the proposed approximation and to verify the theoretical bound in (8). In subsection 4.2, we shall demonstrate the efficacy of the proposed optimization algorithm in reducing $P_{cx}$. All BEPs in all experiments are estimated using Monte-Carlo simulation. The number of trials is selected such that $N \geq \max(5 \times 10^4, (1 - \hat{P}_c)/(0.05^2 \hat{P}_c))$, where $\hat{P}_c$ is the estimated BEP. This selection ensures that the ratio between the standard deviation of the estimate to its mean does not exceed 0.05.

4.1. Quality of the proposed quadratic approximation
In this simulation, we fixed $n$ to 30 while $C$ is changed from 2 to 40. For each value of $C$, 10 Gaussian classification problems are synthetically generated. The distributions of the generated $P(H_i)$ and $\Sigma_i$ are shown in Table 1 (heteroscedastic classification) and $\mu_i$ is generated according to $\mathbb{N}(0, 5I)$. For each classification problem, both the $P_{cx}$ and its corresponding quadratic approximation $Q_y(\phi_0^*, \phi_2^*)$ are estimated using the method of Monte-Carlo. The actual values of $|P_{cx} - Q_y(\phi_0^*, \phi_2^*)|$ as well as the theoretical upper bound in (8) are plotted in Figure 2. We notice that the actual difference saturates at 0.1 with the increase of $C$. Hence, as $C$ increases, $Q_y(\phi_0^*, \phi_2^*)$ should still provide a reasonable approximation to $P_{cx}$. Therefore, it is expected that the performance of the proposed optimization procedure is not much affected by the increase of $C$. In addition, the difference between the theoretical upper bound of $|P_{cx} - Q_y(\phi_0^*, \phi_2^*)|$ and its actual value does not exceed 0.025 since the theoretical upper bound does not exceed 0.125. This indicates the tightness of the derived theoretical upper bound.

4.2. Improvement achieved by the proposed framework
We considered three types of Gaussian classification problems: (1) heteroscedastic (general) Gaussian classification problems; (2) Gaussian classification problems with spherical covariance matrices; i.e. $\Sigma_i = \sigma_i^2 I$; (3) Homoscedastic Gaussian classification problems, $\Sigma_i = \Sigma$.

For each type, 100 synthetic Gaussian classification problems were randomly generated. The distributions of the generated classification parameters are shown in Table 1. The mean vectors were generated differently for each type of Gaussian classification so as to have reasonably small values of BEPs for each type of classification. We allowed $C$ to take the values of 10, 20, 30, and 40 while $n$ was fixed to 30. We employed three LFE methods as initializers for our optimization procedure: the Fisher LDA, the Tubbs method [8], and the Chernoff-based LDA [5]. The quadratic approximation in (10), our objective function, is optimized using the Broyden Fletcher Goldfarb Shanno (BFGS) method. The maximum number of iterations was 70 while the convergence termination criterion was set to $5 \times 10^{-4}$. For each generated classification problem, the following quantities were estimated: (1) $P_{e_{F11}}$, $P_{e_{T11}}$, $P_{e_{C11}}$ are the BEPs obtained after applying the Fisher LDA method, the Tubbs method, the Chernoff-based method without optimization, respectively, (2) $P_{e_{F12}}$, $P_{e_{T12}}$, $P_{e_{C12}}$ are the BEPs obtained after optimizing the transformation matrix corresponding to $P_{e_{F11}}$, $P_{e_{T11}}$, $P_{e_{C11}}$, respectively, and (3) $k_{F1}$, $k_{T1}$, $k_{C1}$ are the corresponding reduced dimensionalities.

Tables 2 lists the average values of the quantities defined...
Table 1. Distribution of the generated parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Covariance type</th>
<th>Generation density</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P(H_i)$</td>
<td>All</td>
<td>Dirichlet; Dir(1, ..., 10)</td>
</tr>
<tr>
<td>$\mu_i$</td>
<td>Homoscedastic</td>
<td>Multivariate normal; $N(0, 0.511)$</td>
</tr>
<tr>
<td></td>
<td>Spherical</td>
<td>Multivariate normal; $N(0, 0.511)$</td>
</tr>
<tr>
<td>$\Sigma_i$</td>
<td>Homoscedastic</td>
<td>Wishart (50 degrees of freedom)</td>
</tr>
<tr>
<td></td>
<td>Spherical, $\Sigma_i = \Sigma_0$</td>
<td>Wishart (50 degrees of freedom)</td>
</tr>
<tr>
<td></td>
<td>Heteroscedastic</td>
<td>Wishart (50 degrees of freedom)</td>
</tr>
</tbody>
</table>

Table 2. Homoscedastic Gaussian classification.

<table>
<thead>
<tr>
<th>C</th>
<th>$P_{Pe_{F11}}$</th>
<th>$P_{Pe_{F12}}$</th>
<th>$k_{F1}$</th>
<th>$P_{Pe_{Tu1}}$</th>
<th>$P_{Pe_{Tu2}}$</th>
<th>$k_{Tu}$</th>
<th>$P_{Pe_{Ch1}}$</th>
<th>$P_{Pe_{Ch2}}$</th>
<th>$k_{Ch}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.0041</td>
<td>0.0034</td>
<td>6.42</td>
<td>0.0050</td>
<td>0.0044</td>
<td>6.48</td>
<td>0.0050</td>
<td>0.0044</td>
<td>6.44</td>
</tr>
<tr>
<td>20</td>
<td>0.0242</td>
<td>0.0198</td>
<td>7.00</td>
<td>0.0275</td>
<td>0.0219</td>
<td>7.00</td>
<td>0.0241</td>
<td>0.0198</td>
<td>7.00</td>
</tr>
<tr>
<td>30</td>
<td>0.0565</td>
<td>0.0404</td>
<td>7.00</td>
<td>0.0626</td>
<td>0.0447</td>
<td>7.00</td>
<td>0.0561</td>
<td>0.0404</td>
<td>7.00</td>
</tr>
<tr>
<td>40</td>
<td>0.0635</td>
<td>0.0535</td>
<td>7.00</td>
<td>0.0762</td>
<td>0.0565</td>
<td>7.00</td>
<td>0.0635</td>
<td>0.0535</td>
<td>7.00</td>
</tr>
</tbody>
</table>

above for homoscedastic Gaussian classification over the 100 generated problems. The reduced dimensionality is determined as the one providing the minimum BEP (before optimization) within the domain from 1 to 7. Generally, the three methods have very comparable classification performance before and after optimization. This is consistent with two facts for homoscedastic Gaussian classification problems: (a) the Chernoff method reduces to the Fisher LDA [5] and (b) when $k \geq k_{\text{min}}$, both the Fisher and the Tubbs methods are equivalent since the left singular vectors of $\Sigma$ in (5) are related to the Fisher eigenvectors by a nonsingular transformation. Therefore, it is very likely that these LFE methods will perform close to optimum when $k < k_{\text{min}}$ and there is no significant improvement in the classification performances.

The experiments were repeated for the other two types of Gaussian classification problems in Table 3 and Table 4. The proposed optimization algorithm provides significant reduction in the BEP as compared to the homoscedastic classification. This is expected since, with probability one, $k_{\text{min}} = n$. Hence, $y$ in the three LFE methods will not be a sufficient statistic for $x$ and it is very likely that none of the three methods will be optimal. In addition, the amount of reduction in the BEP increases as $C$ increases, or equivalently, when the classification task becomes harder. This adds to the strength of our proposed optimization algorithm. It also means that as $C$ increases, the three LFE methods become less optimal and there is more need for optimizing $P_{xy}$ w.r.t. A. Further, the BEPs of the three methods after optimization are very comparable to each other. That is, the final obtained classification performance after optimization is somewhat insensitive to the initializer. We believe that this property is very desirable in many signal classification problems since it is difficult to know which LFE method will yield the best classification performance beforehand.

Finally, the selected dimensionality in all Table 2, 3, and 4 was either exactly 7 or very close to it. This is consistent with the intuition of the monotonic decreasing behaviour of the $P_{xy}$ versus $k$ when the class conditional densities are exactly known.

5. CONCLUSIONS

In this paper, we have proposed a novel framework for improving the performance of LFE algorithms, characterized by the BEP after dimensionality reduction. The improvement has been validated experimentally for three popular LFE algorithms. Based on the performed simulations, it was concluded that, unless the transformed feature vector is a sufficient statistic for the original pattern vector, no single LFE method is guaranteed to have the best classification performance for all Gaussian classification problems. Though the focus in this paper has been on classification problems with known conditional densities, we believe the same conclusion still applies when the class conditional densities are estimated from given training data. This case will be further investigated in future publications.

References