Development of Kernel Fisher Discriminant Model Using The Cross-Entropy Method

Budi Santosa
Department of Industrial Engineering
Institut Teknologi Sepuluh Nopember (ITS)
Surabaya, Indonesia
budi_s@ie.its.ac.id

Andiek Sunarto
Department of Industrial Engineering
Institut Teknologi Sepuluh Nopember (ITS)
Surabaya, Indonesia
andiek76@ie.its.ac.id

Abstract—In this paper, the cross-entropy (CE) method is proposed to solve non-linear discriminant analysis or Kernel Fisher discriminant (CE-KFD) analysis. CE through certain steps can find the optimal or near optimal solution with a fast rate of convergence for optimization problem. While, KFD is to solve problem of Fisher’s linear discriminant in a kernel feature space \( F \) by maximizing between class variance and minimizing within class variance. Through the numerical experiments, we found that CE-KFD demonstrates the high accuracy of the results compared to the traditional methods, Fisher LDA and kernel Fisher (KFD) with eigen decomposition method.

Keywords—accuracy, cross entropy, discriminant analysis, eigen decomposition, kernel method

I. INTRODUCTION

Kernel Fisher Discriminant Analysis is an algorithm which is used to overcome non-linearity involved at classification. It mapped the two sets of data which cannot be separated linearly into a potentially much higher dimensional feature space \( F \). Kernel Fisher discriminant (KFD) analysis were applied successfully for various fields, for instance in application to speech impediment therapy [2]. According to our knowledge, there are two approaches to solve KFD. First, using eigen decomposition method and second, using a mathematical programming approach [6].

This paper proposed the cross-entropy method to solve the KFD problem. The aim is to have better or the same accuracy and computing time. This refers to some researches in which showed the surprising results when applying the CE method to overcome the optimization problem. The CE method is one of the most exciting development in stochastic optimization and simulation in recent years. It is applied mostly for rare-event simulation, combinatorial problem, and multi-extremal optimization. Many research have been done since cross-entropy were launched for the first time, such as the cross-entropy method for clustering and vector quantization [4], portfolio selection [7], continous multi extremal optimization ([3], [9]), dual lagrange SVM [11], max-cut problem [5].

The experiment shows that CE method gave the competitive results in accuracy compared to the kernel eigen decomposition method and LDA. These three methods are applied on six real world data sets: Breast Cancer, Pima India, WDBC, Sonar, Bupa, and Ionosphere.

The paper is organized as follows. The second section reviews theory, model, analysis, and implementation of Fisher linear discriminant analysis, KFD, cross-entropy method, and the proposed algorithm. In the third section is shown the results of the experiments and the last section concludes and discusses overall of the research.

II. KERNEL FISHER DISCRIMINANT ANALYSIS

Kernel Fisher discriminant analysis is the extended version of Fisher linear discriminant. Fisher linear discriminant analysis (LDA) aims to separate two classes of data linearly through discriminant function. Let \( X_1 = \{x_{1,1}, \ldots, x_{1,l}\} \) and \( X_2 = \{x_{2,1}, \ldots, x_{2,l}\} \) be samples from two different classes and with some abuse of notation \( X = X_1 \cup X_2 = \{x_{1,1}, \ldots, x_{l}\} \). \( l \) is number of sample \( X_1 \) and \( l \) is number of sample \( X_2 \). LDA should find the projection vector \( w \), so discriminant function \( wx + b \), where \( b \) is offset vector, will guarantee term this below maximized.

\[
J(w) = \frac{w^T S_B w}{w^T S_w w} \tag{1}
\]

where

\[
S_B := (m_1 - m_2)(m_1 - m_2)^T \quad \text{and} \quad \tag{2}
\]

\[
S_w := \sum_{i=1,2, l} \sum_{j=1}^l (x - m_i)(x - m_j)^T \tag{3}
\]

\(S_B \) and \( S_w \) respectively measure the between and within class scatter matrices respectively and \( m_i \) is mean vector of samples \( i \) which defined by \( m_i := \frac{1}{l} \sum_{j=1}^l x_j^i \). The idea behind maximizing \( J(w) \) is to find a direction which
maximizes the projected class means (the numerator) while minimizing the classes variance in this direction (the denominator). LDA has proven very powerful to separate two classes of data sets which can be separated linearly. One reason is certainly that a linear model is rather robust against noise and most likely will not overfit. Unfortunately, LDA can not be applied to non-linear cases, even the more part of real world data sets are non-linear. However, kernel Fisher discriminant (KFD) is proposed to overcome this problem. KFD mapped nonlinear data sets in input or original space into a potentially much higher dimensional feature space $F$, such that the data can be separated linearly. In feature space LDA can be applied, such that implicitly yielding a non-linear discriminant in input space. Let $\Phi$ be a non-linear mapping to some feature space $F$. To find the linear discriminant in $F$ we need to maximize

$$J(w) = \frac{w^T S_B^\phi w}{w^T S_W^\phi w}$$

(4)

where $w \in F$ and $S_B^\phi$ and $S_W^\phi$ are the corresponding matrices in $F$, i.e.

$$S_B^\phi := (m_1^\phi - m_2^\phi)(m_1^\phi - m_2^\phi)^T$$

and

$$S_W^\phi := \sum_{i=1}^{l_1} (\Phi(x_i) - m_1^\phi)(\Phi(x_i) - m_2^\phi)^T$$

(5)

with

$$m_i^\phi := \frac{1}{l_i} \sum_{j=1}^{l_i} \Phi(x_j)$$

(7)

Mapping into the higher dimensional feature space will increase the complexity of the function class matters [6].

Fortunately, for certain feature space $F$ and corresponding mapping $\Phi$ there is a highly effective trick for computing scalar products in feature spaces using kernel functions [6]. Instead of mapping the data explicitly we seek a formulation of the algorithm which uses only dot-product ($\Phi(x_i) \cdot \Phi(x_j)$) of the training patterns. As we are then able to compute these dot-products efficiently we can solve the original problem without ever mapping explicitly to $F$. Possible choice for $k$ which have proven useful are Gaussian RBF, $k(x,y) = \exp(-\|x - y\|^2 / c)$, or polynomial kernels, $k(x,y) = (x \cdot y)^d$, for some positive constant $c$ and $d$ respectively.

To find Fisher’s discriminant in the feature space $F$, we first need a formulation (5) in terms of only dot products of input patterns which we then replace by some kernel function. Define

$$w := \sum_{i=1}^{l} \alpha_i \phi(x_i)$$

(9)

Using the expansion (9) and the definition $m^\phi$ we get

$$w^T m_i^\phi = \frac{1}{l_i} \sum_{j=1}^{l_i} \alpha_j k(x_j, x_i^\phi)$$

$$= \alpha^T M_i$$

(10)

where we defined

$$(M_i)_j := \frac{1}{l_i} \sum_{k=1}^{l_i} k(x_j, x_k^\phi)$$

and replaced the dot products by kernel function. Now consider the numerator of (5). Be using the definition of $S_B^\phi$ and (10) it can be rewritten as

$$w^T S_B^\phi w = \alpha^T M \alpha$$

(11)

where $M := (M_1 - M_2)(M_1 - M_2)^T$. Considering the denominator, using (9), the definition of $m^\phi$ and a similar transformation as in (11) we find

$$w^T S_W^\phi w = \alpha^T N \alpha$$

(12)

where we set $N := \sum_{j=1}^{l} K_j (I - 1_{l_j}) K_j^T$, $K_j$ is a $l \times l_j$ matrix with $(K_j)_{kn} := k(x_{kn}, x_{kn}^\phi)$ this is kernel matrix for class $j$), $I$ is identity and $1_{l_j}$ the matrix with all entries $1/l_j$. Combining (11) and (12) we can find Fisher’s linear discriminant in $F$ by maximizing

$$J(\alpha) = \frac{\alpha^T M \alpha}{\alpha^T N \alpha}$$

(13)

This problem can be solved (analogously to the algorithm in the input space) by some optimization method, such as finding eigen vector, gradient descent, and the other. The projection of a new pattern $x$ onto $w$ is given by

$$(w \cdot \Phi(x)) = \sum_{i=1}^{l} \alpha_i k(x, x_i)$$

(14)

III. THE CROSS ENTROPY METHOD

The cross entropy method is a new approach of the stochastic optimization algorithm. The principles of the CE method is provided by Rubinstein [8], Rubinstein and Kroese [10], and De Boer et al. [1]. The basic idea of the CE method is to transform the original optimization problem to the associated stochastic optimization problem, and handle the stochastic problem efficiently by an adaptive sampling algorithm. The CE method involve an iterative procedure where each iteration can be broken down into two phase:

1. Generate a random data sample (trajectories, vectors, etc.) according to a specified mechanism.
2. Update the parameters of the random mechanism based on the data to produce a “better” sample in the next iteration.
Suppose we wish to minimize some cost function $S(z)$ over all $z$ in some set $Z$. Let us denote the minimum by $\gamma^*$, thus
$$\gamma^* = \min_{z \in Z} S(z) \quad (15)$$
We randomize our deterministic problem by defining a family of auxiliary pdfs $\{f^{(\ast)}(\cdot; v), v \in V\}$, and associated with eq. (15) the following estimation problem for a given scalar $\gamma$
$$P_u(S(Z) \leq \gamma) = E_u[I_{\{S(Z)\leq \gamma\}}]$$
where $u$ is some known (initial) parameter. We consider the event “cost is low” to be rare event $I_{\{S(Z) \leq \gamma\}}$ of interest. To estimate the event, the CE method generates a sequence of tuples $\{(\gamma^*_1, v_1)\}$, that converge (with high probability) to small neighborhood of the optimal tuple $\{(\gamma^*, v^*)\}$, where $\gamma^*$ is the solution of the program (15), and $v^*$ is a pdf that emphasize values in $Z$ with low cost. We note that typically the optimal $v^*$ is degenerated as it concentrates on the optimal solution (or small neighborhood thereof). Let $\rho$ denote the fraction of the best samples used to find the threshold $\gamma$. The process that is based on sampled data is termed the stochastic counterpart since it is based on stochastic samples of data. The number of samples in each stage of the stochastic counterpart is denoted by $N$, which is a predefined parameter. The following is a standart CE procedure for minimization borrowed from de-Boer et al. [1]. We initialize $\hat{v}_0 = \bar{v}_0 = u$ and choose a not very small $\rho$, say $10^{-2} \leq \rho$ . We then proceed iteratively as follows:

1. Adaptive updating of $\gamma_i$.
   A simple estimator $\hat{\gamma}_i$ of $\gamma_i$ can be obtained by taking random sample $Z(l), ..., Z(N)$ from the pdfs $f^{(\ast)}(\cdot; v_{\gamma_i})$, calculating the performance $S(Z(l))$ for all $l$, ordering them from smallest to biggest as $S(l), ..., S(N)$ and finally evaluating the $100\%$ sample percentile as $\hat{\gamma}_i = S_{\{l|\rho N\}}$

2. Adaptive updating of $v_i$.
   For a fixed $\gamma_i$ and $v_{\gamma_i}$, derive $v_i$ from the solution of the program:
   $$\max_{v} D(v) = \max_v E_{v_{\gamma_i}}^{(1)} I_{\{S(z) \leq \gamma\}} \log f(Z; v) \quad (16)$$
   The stochastic counterpart of (16) is follows: $\hat{\gamma}_i$ and $\hat{\gamma}_{i-1}$, derive $\hat{v}_i$ from the following program:
   $$\max_{v} \hat{D}(v) = \max_v \frac{1}{N} \sum_{l=1}^{N} I_{\{S(Z(l)) \leq \hat{\gamma}_i\}} \log f(Z(l); v) \quad (17)$$
   In this paper we assume that $f$ belongs to Gaussian family. In our case $Z \in [0, C]^n$ and $v$ is a $n$ dimensional vector of numbers between 0 and 1, and $C$ is constant defined by users. The constant $C$ is upper bound of lagrange multiplier as which we seek for. The update formula of the $k^{th}$ element in $v$ (eq. (17)) in this case simply becomes
   $$\hat{v}_i(k) = \frac{\sum_{l=1}^{N} I_{\{S(Z(l)) \leq \hat{\gamma}_i\}} I_{\{Z(l) \leq \hat{\gamma}_i\}}}{\sum_{l=1}^{N} I_{\{S(Z(l)) \leq \hat{\gamma}_i\}}} \quad (18)$$
   To simplify eq. (18), we can use the following smoothed version that provided by Kroese et al [3], Santosa [11]
   $$\hat{v}_i = \beta \hat{v}_{i-1} + (1 - \beta) \hat{\gamma}_{i-1} \quad (19)$$
   Where $\hat{v}_i$ is the parameter vector obtained from the solution of (17), and $\beta$ is a smoothing parameter, with $0.7 \leq \beta \leq 1$. The CE optimization algorithm can be found in [12]. It is found empirically that the CE method is robust with respect to the choice of its parameter $N, \rho$, and $\beta$, as long as $\rho$ is not too small, $\beta < 1$, and $N$ is large enough. Typically those parameters satisfy that $0.01 \leq \rho \leq 0.1$, $0.5 \leq \beta \leq 0.9$, and $N \geq 3n$, where $n$ is the number of parameter.

IV. THE PROPOSED APPROACH: KFD BY CROSS ENTROPY METHOD

In this section we apply the CE method to solve the optimization program of some classifier problem, say Fisher linear discriminant and kernel Fisher discriminant. Application The CE method to kernel Fisher discriminant is simarily with its application to Fisher linear discriminant, the difference is it done in feature space. For instance, we will explain it as follows. First step, we define kernel function $k$. For all training data sets, we calculate $k$ value. According eq. (11) and (12), we define $M$ and $N$ value, and thus we obtain the score function $J(\alpha)$, where $\alpha$ is the projection vector in feature space. Our aim is to find $\alpha$ in feature space by generating some random vectors $\alpha$ until $J(\alpha)$ is maximized Similary the CE method to Fisher linear discriminant, we initialize We initialize $\hat{v}_0 = \mu = 0$, $\sigma$ is 1 while the components of vector $w$ must be lied between 0 and 1. Initialize N large enough, say 100, $\rho = 0.1$, and $\beta = 0.7$. Epsilon is defined very small, say $10^{-6}$. The complete steps is provided by algorithm 3 as follows.

Algorithm 4 The CE method to Kernel Fisher Discriminant.
1. generate the $N$-random sample vectors in feature space as initial $\mu$ and $\sigma$ parameters
2. this step has been done until $\sigma$ reaches epsilon do to steps 3 to 6.
3. calculate the score of $\alpha$ by inputing to the score function $J(\alpha)$ (eq. (13)).
4. sort these values and select the $\mu$ values providing $\rho\%$ percentile biggest scores of $J(\alpha)$.
5. compute the new means and deviation standarts of $\rho\%$-percentile of the selected $\mu$ values
6. update and smoothen the $\mu$ and $\sigma$ parameter which were obtained on the $6^{th}$ step results. Use eq. (19) by associating $\mu$ and $\sigma$ as $\hat{v}_i$. Return to the first step.

In the following classifier is used to test new data points
$$f(x) = \text{sign} \left( \sum_{i=1}^{n} v_i \alpha_i K(x, x_i) \right)$$
V. EXPERIMENTS AND RESULTS

To evaluate the performance of our approach, we apply the CE-KFD along with KFD with eigen and LDA on six real world data sets taken from the UCI repository [22]. Table 1 shows the characteristic of these data sets with differences in features and patterns.

<table>
<thead>
<tr>
<th>No.</th>
<th>Name</th>
<th>#Features</th>
<th>#Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Breast cancer</td>
<td>9</td>
<td>683</td>
</tr>
<tr>
<td>2.</td>
<td>Ionosphere</td>
<td>34</td>
<td>351</td>
</tr>
<tr>
<td>3.</td>
<td>Pima</td>
<td>8</td>
<td>768</td>
</tr>
<tr>
<td>4.</td>
<td>WDBC</td>
<td>30</td>
<td>569</td>
</tr>
<tr>
<td>5.</td>
<td>Bupa</td>
<td>6</td>
<td>345</td>
</tr>
<tr>
<td>6.</td>
<td>Sonar</td>
<td>60</td>
<td>208</td>
</tr>
</tbody>
</table>

The experimental set-up can be explained as follows. The data sets is divided into training and testing sets where the ratio is 60 : 40. The experiments were run 10 times on 10 different training and testing sets. The generalization error is computed by taking the average of 10-runs misclassification values. The complete results are summarized in Table 2.

The results show that the CE-KFD is superior in terms of accuracy to KFD with eigen decomposition in Breast Cancer, Bupa and Sonar data sets, yet with computational time higher. Only on Ionosphere, kernel eigen performed better than CE-KFD. In this case, LDA failed because the pooled covariance matrix on training set is not positive definite. While in Pima and WDBC data, LDA provided the best accuracy.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Breast Cancer</td>
<td>Eigen</td>
<td>0.038</td>
<td>1.968</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CE</td>
<td><strong>0.031</strong></td>
<td><strong>1.566</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA</td>
<td>0.045</td>
<td>0.005</td>
</tr>
<tr>
<td>2.</td>
<td>Ionosphere</td>
<td>Eigen</td>
<td><strong>0.080</strong></td>
<td><strong>0.24</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CE</td>
<td>0.091</td>
<td>1.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>failed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Pima</td>
<td>Eigen</td>
<td>0.313</td>
<td><strong>2.792</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CE</td>
<td><strong>0.266</strong></td>
<td>8.588</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA</td>
<td>0.2373</td>
<td>0.005</td>
</tr>
<tr>
<td>4.</td>
<td>WDBC</td>
<td>Eigen</td>
<td>0.097</td>
<td>1.049</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CE</td>
<td>0.073</td>
<td>1.064</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA</td>
<td><strong>0.037</strong></td>
<td><strong>0.002</strong></td>
</tr>
<tr>
<td>5.</td>
<td>Bupa</td>
<td>Eigen</td>
<td>0.3400</td>
<td>0.245</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CE</td>
<td><strong>0.3330</strong></td>
<td><strong>6.786</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA</td>
<td><strong>0.3700</strong></td>
<td><strong>0.003</strong></td>
</tr>
<tr>
<td>6.</td>
<td>Sonar</td>
<td>Eigen</td>
<td>0.25</td>
<td><strong>0.059</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CE</td>
<td><strong>0.24</strong></td>
<td>2.558</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LDA</td>
<td>0.29</td>
<td>0.005</td>
</tr>
</tbody>
</table>

VI. CONCLUSION AND DISCUSSION

In this paper we showed how KFD can be solved not only by analytical or deterministic method (eigen decomposition method) but also stochastic method (cross entropy method). Some advantages of applying the CE method to solve KFD algorithm is: 1) it gives the new alternative method to optimize the classifier function of KFD; 2) CE-KFD does not face with the problem of positive definiteness of the covariance matrix as in LDA; 3) For the most of the experimented data sets, the accuracy of CE-KFD is promising.

REFERENCES