Feature Extraction for One-Class Classification Problems: Enhancements to Biased Discriminant Analysis

Nojun Kwak and Jiyong Oh

Nojun Kwak is an assistant professor at the Division of Electrical & Computer Engineering, Ajou University, Suwon, Korea.

Jiyong Oh is a Ph.D. student at the School of Electrical Engineering and Computer Science, Seoul National Univ., Seoul, Korea.

April 29, 2008

Abstract

In many one-class classification problems such as face detection and object verification, the conventional linear discriminant analysis sometimes fails because it makes an inappropriate assumption on negative samples that they are distributed according to a Gaussian distribution. In addition, it sometimes can not extract sufficient number of features because it merely makes use of the mean value of each class. In order to resolve these problems, in this paper, we extend the biased discriminant analysis (BDA) which was originally developed for one-class classification problems. The BDA makes no assumption on the distribution of negative samples and tries to separate each negative sample as far away from the center of positive samples as possible. The first extension uses a saturation technique to suppress the influence of the samples which are located far away from the decision boundary. The second one utilizes the L1 norm instead of the L2 norm. Also we present a method to extend BDA and its variants to multi-class classification problems. Our approach is considered useful in the sense that without much complexity, it successfully reduces the negative effect of negative samples which are far away from the center of positive samples, resulting in better classification performances. We have applied the proposed methods to several classification problems and compared the performance with conventional methods.

Keywords

Classification, one-class, one-against-rest, BDA.

I. Introduction

Classification has long been a classical problem in pattern recognition and machine learning society. In classification problems with large number of input variables, dimensionality reduction methods are typically used to simplify the classifier system without degrading performances. Among these, *linear dimensionality reduction* (LDR) techniques have been extensively studied by many researchers and the *linear discriminant analysis* (LDA) [1] is the most popular and successful.

LDA tries to find a transformation which maximizes the ratio of the between class scatter and the within class scatter. It is known to be optimal for the cases where each class is distributed according to a uni-modal Gaussian distribution with the same covariance matrix [1].

However, in many classification problems, we are sometimes interested in discriminating only one class from the other samples. This problem is called as the *one-against-rest* problem or just as the *one-class* classification problem. Face detection and object verification problems are typical examples. In these problems, positive samples can be assumed to be uni-modal, but no assumption can be made on the distribution of negative samples. For these problems, a simple LDA approach may produce misleading results because it tries to cluster negative samples, which is of little use in increasing discriminating power.

In addition, for these problems, the conventional LDA can produce only one feature because it merely tries to separate the mean values of positive and negative samples as much as possible. This scarcity of extracted features can be mitigated by several techniques [2] [3]. In [3], the Chernoff criterion is incorporated into the original structure of LDA to make use of the covariance differences among different classes as well as the mean differences. However, it still assumes that all classes have uni-modal distributions with different means and covariances. Thus, it does not fit to the problems where one can not make any assumption on the distribution of negative samples.

Several approaches can be considered to cope with multi-modal characteristics of negative samples. The simplest one is to regard each negative sample as belonging to a distinct class. Thus, in this approach, if the number of negative samples is n_y, it becomes an $(n_y + 1)$-class classification problem. On the other hand, in [4], each class is subdivided into multiple subclasses by unsupervised clustering techniques and then LDA is applied to discriminate the subclasses. However as stated in [5], for one-against-rest problems, *any constraint put on the negative samples other than "stay away from the positive" is unnecessary and misleading.*

For one-against-rest problems, BDA (biased discriminant analysis) was proposed in [5]. In this approach, the positive and negative samples are treated differently and the only criterion put on the negative samples is *"stay away from the positive"*. Therefore, it tries to find a transformation that concentrates the positive samples around their center and separate the negative samples far away from the center. Though it is well suited to one-against-rest problems, its result may sometimes depend too much on a few negative samples which are very far away from the positive center because it tries to maximize an objective function based on the L2 norm.

In this paper, we propose two methods to cope with this problem. The first one is to use a saturation technique; by setting an upper bound on the distance of a negative sample from the positive center we can restrict the dominance of negative samples which are far away from the decision boundary. In the second method, instead of using the L2 norm, the L1 norm is used. In this way, the objective function increases linearly with the distance of a negative sample from the positive center rather than quadratically. In addition, the method which combines the saturation scheme with L1 norm optimization is also presented. Finally, a simple method is presented in order to extend BDA which was originally devised for one-class classification problems to multi-class classification problems.

This paper is organized as follows. In Section II, we briefly overview the conventional BDA. Two extensions of BDA are proposed in Section III. Section IV shows experimental results and an approach to extend BDA approaches to multi-class classification problems is also presented in due course. Finally, conclusions follow in Section V.

II. A brief overview of BDA

In BDA, which was originally proposed for one-class classification problems, positive samples and negative samples are treated differently. While the positive samples are assumed to be generated by Gaussian distribution, no assumption is made on the distribution of negative samples. In order to achieve good discriminating power, BDA tries to find a transformation which concentrates the positive samples around their center while separating the negative samples far away from the center. This is formulated to find M projection vectors $\{\boldsymbol{w}_n\}_{n=1}^{M}$ that maximizes the following objective function:

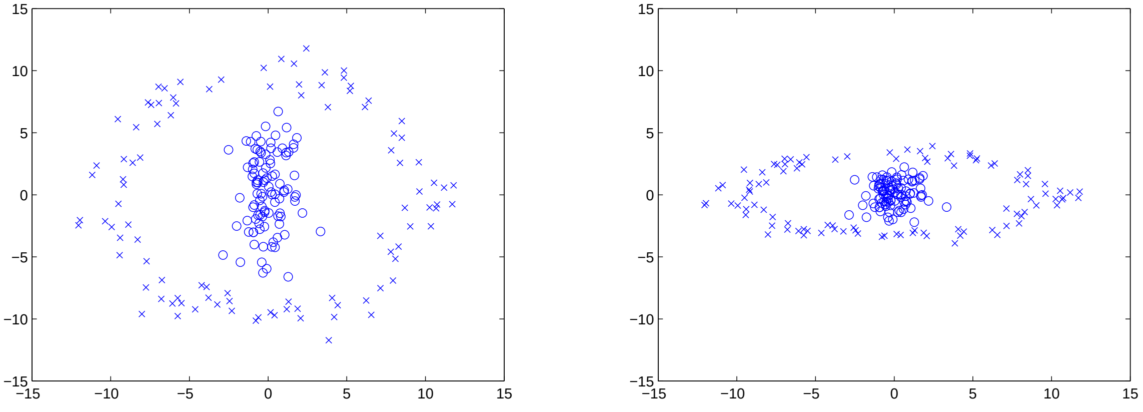
$$W^* = \underset{W}{\operatorname{argmax}} \frac{|W^T S_y W|}{|W^T S_x W|}. \tag{1}$$

Here the n-th column of W corresponds to $\boldsymbol{w}_n$. The matrices S_x and S_y are the positive and negative scatter matrices respectively which are defined as:

$$S_x = \sum_{i=1}^{n_x} (\boldsymbol{x}_i - \boldsymbol{m_x})(\boldsymbol{x}_i - \boldsymbol{m_x})^T \tag{2}$$

$$S_y = \sum_{i=1}^{n_y} (\boldsymbol{y}_i - \boldsymbol{m_x})(\boldsymbol{y}_i - \boldsymbol{m_x})^T, \tag{3}$$

where $\{\boldsymbol{x}_i\}_{i=1}^{n_x}$ and $\{\boldsymbol{y}_i\}_{i=1}^{n_y}$ denote the positive and negative samples respectively and $\boldsymbol{m_x}$ is the mean vector of the positive samples.



(a)Before sphering (Original)

(b)After sphering

Fig. 1. Data distribution before and after sphering process

In solving the optimization problem (1), the following two-step procedure is generally used.

1. Sphering: Solve the eigenvalue decomposition problem $S_x U = U\Lambda_1$. Then, scale each column $\boldsymbol{u}_n$ of U to make $\{\hat{x}_{in} = \hat{\boldsymbol{u}}_n^T(\boldsymbol{x}_i - \boldsymbol{m}_{\boldsymbol{x}})\}_{i=1}^{n_x}$ have unit variance. Here $\hat{\boldsymbol{u}}_n$ is a scaled version of $\boldsymbol{u}_n$. If $||\boldsymbol{u}_n|| = 1$, this is equivalent to setting $\hat{\boldsymbol{u}}_n = \boldsymbol{u}_n/\sqrt{\lambda_{1n}}$, where λ_{1n} denotes the n-th eigenvalue of S_x.
2. Maximization: Find M weight vectors $\{\boldsymbol{v}_n\}_{n=1}^{M}$ that maximizes the following objective function:

$$V = \underset{W}{\operatorname{argmax}}\, |W^T \hat{S}_y W|, \quad \text{subject to} \quad W^T W = I, \tag{4}$$

where $\hat{S}_y = \hat{U}^T S_y \hat{U}$. Then output $W = \hat{U}V$. Here, $\hat{U} = [\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2, \cdots]$.

This procedure is illustrated in Fig. 1. In the figure, the circles represent 100 positive samples and × marks represent 100 negative samples. The positive samples were generated from normal distribution with zero mean and the covariance matrix of $[1, 0; 0, 3]$. The average distance of the negative samples from the origin was set to 10 with a variance of 1, and their orientations were chosen randomly from a uniform distribution.

In the original input space (Fig. 1(a)), we can see that the horizontal direction has the most discriminating power. After the sphering process, in Fig. 1(b), positive samples were rescaled to have unit variance in all orientations. Applying the maximization step on Fig. 1(b), the horizontal direction will be chosen as the first feature as expected.

Note that if S_x is nonsingular, the sphering process is always possible. On the other hand, if S_x is singular, the regularization techniques [6] [7] [8] [5] [9] can be used to make the sphering process possible. In the experiments in Section IV, we use the following regularization technique for the BDA approaches [9] in which S_x is modified to S'_x to avoid singularity.

$$S'_x = S_x + \alpha I_d. \tag{5}$$

Here a small scalar $\alpha(> 0)$ is a regularization parameter and I_d denotes a d-dimensional identity matrix.

From now on, we will assume that all the datasets are preprocessed by the sphering process such that the positive samples have a Gaussian distribution with zero mean and identity covariance matrix. In addition, V will be denoted as W hereafter.

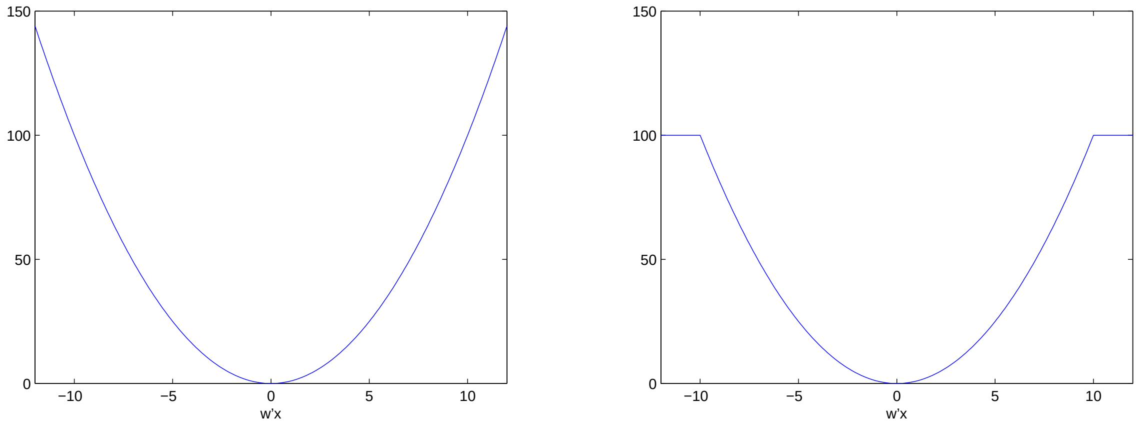
III. Enhancements to BDA

A. Shaping the objective function

In BDA, the distances of negative samples from the positive mean are usually greater compared to those of positive samples and the positive and negative samples can be separated based on these distances (e.g., see Fig. 1.).

If we assume that positive samples have Gaussian distribution, they will have unit variance in all orientations after the sphering operation. In an one-dimensional input space, 99% of the positive samples will be within the ball of radius 3, and if a sample is outside this ball, we can say with 99% confidence that this sample is not positive and thus negative. Therefore, it does not make a big difference in discriminating power whether a negative sample is projected at a point whose distance from the center of positive samples is 10 or 100.

However, the objective function (4) is quadratic and it may be very sensitive to samples with large norm. This can cause a serious problem when most of the negative samples are near the center of the positive samples and only a few are located very far from the center. In this case, the objective function is dominated by these few and the discriminating power may degrade. For example, if we add a negative sample at location (0, 500) to the problem shown in Fig. 1, this point will dominate the objective function and the transformation $\boldsymbol{w}$ that maximizes the objective function will be close to the vertical line. In this case, we



(a)Original

(b)Modified ($\gamma = 10$)

Fig. 2. Objective functions

can see in Fig. 1 (b) that most of the negative samples projected along $\boldsymbol{w}$ will be close to 0 in the range of [-5, 5] while positive samples will be projected in the range of [-3, 3] and classifying these samples will be difficult.

In order to resolve this problem, instead of using a quadratic objective function (4), we would like to modify the objective function to reduce the effect of the samples which have large norms. The first thing we can think of is to set an upper bound on the norm and change the objective function as the following:

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{\mathrm{argmax}} \sum_{i=1}^{n_y} \min((\boldsymbol{w}^T \boldsymbol{y}_i)^2, \gamma^2) \quad \text{subject to} \quad ||\boldsymbol{w}|| = 1. \tag{6}$$

Here γ is a saturation constant. Figure 2 shows the original object function and the modified one with $\gamma = 10$.

Because the minimization operation in (6) is nonlinear, it is difficult to find the optimal vector $\boldsymbol{w}^*$. In order to make the problem easier, instead of applying the minimization operation after the projection, it is applied before the projection and the objective function is modified as the following:

$$\boldsymbol{w}^* = \mathrm{argmax}_{\boldsymbol{w}} \textstyle\sum_{i=1}^{n_y} (\boldsymbol{w}^T \boldsymbol{z}_i)^2 \quad \text{where} \quad \boldsymbol{z}_i = \frac{\boldsymbol{y}_i}{||\boldsymbol{y}_i||} \min(||\boldsymbol{y}_i||, \gamma) \tag{7}$$
$$\text{subject to} \quad ||\boldsymbol{w}|| = 1.$$

This is equivalent to the following eigenvalue decomposition problem

$$W^* = \underset{W}{\operatorname{argmax}} |W^T S_z W| \quad \text{subject to} \quad W^T W = I, \tag{8}$$

where $S_z = \sum_{i=1}^{n_y} (\boldsymbol{z}_i - \boldsymbol{m_x})(\boldsymbol{z}_i - \boldsymbol{m_x})^T$, and this problem can easily be solved. From now on, we will refer to this method as the *Saturated-BDA* or *SBDA*.

The minimization operation in (8) can be regarded as setting a ball in the hyperplane with a radius γ and pulling down the negative samples outside this ball onto the surface of the ball. If γ is too large, most of the samples will be within the ball and the solution of SBDA will almost be the same as that of the conventional BDA. On the other hand, if γ is too small, many samples will be outside the ball and these will be pulled down to the surface of the ball leading to a distorted solution. As will be discussed in the experiments in Section IV, we recommend the value of γ to be 1 to 10.

This approach is easy to apply without adding much complexity to the conventional BDA. However, this approach does not give the optimal solution for (6) but an approximate solution. We can show that the solution of (8) gives a lower bound to a solution of (6) as follows.

$$\begin{aligned} \min((\boldsymbol{w}^T\boldsymbol{y}_i)^2, \gamma^2) &= [\min(|\boldsymbol{w}^T\boldsymbol{y}_i|, \gamma)]^2 = [(\boldsymbol{w}^T\boldsymbol{y}_i)\min(1, \frac{\gamma}{|\boldsymbol{w}^T\boldsymbol{y}_i|})]^2 \\ &\geq [(\boldsymbol{w}^T\boldsymbol{y}_i)\min(1, \frac{\gamma}{||\boldsymbol{w}||\;||\boldsymbol{y}_i||})]^2 \\ &= [\boldsymbol{w}^T[\boldsymbol{y}_i \min(1, \frac{\gamma}{||\boldsymbol{y}_i||})]]^2 = (\boldsymbol{w}^T\boldsymbol{z}_i)^2. \end{aligned} \tag{9}$$

The equality holds when $\boldsymbol{y}_i$ and $\boldsymbol{w}$ are parallel. As the number of negative samples outside the ball which produce a large angle with the solution vector $\boldsymbol{w}$ increases, difference between the solutions of (6) and (8) increases.

Note that BDA is not invariant to rotations in general. However, in case where the positive mean is located at the origin, BDA is invariant to rotations. Likewise, if the positive mean is at the origin, SBDA is invariant to rotations. In other words, if we rotate the original data with a rotation matrix R whose columns are orthonormal vectors, the solution W_o of SBDA on the original data and W_r which is the solution of SBDA on the rotated data will be related by $W_o = W_r R$.

B. L1-BDA

From a statistical point of view, it is well known that the methods based on the L1 norm are more robust to the samples, which have a large norm compared to the L2 norm [10]. Therefore, an objective function based on the L1 norm will better reduce the effect of negative samples which have a large norm. This L1 optimization problem can be defined as follows:

$$\boldsymbol{w}^{*} = \underset{\boldsymbol{w}}{\operatorname{argmax}} \sum_{i=1}^{n_y} |\boldsymbol{w}^T \boldsymbol{y}_i| \quad \text{subject to} \quad ||\boldsymbol{w}|| = 1. \tag{10}$$

In order to find the solution vector $\boldsymbol{w}^{\star}$, we replace the 'maximization' step of the conventional BDA (Step 2 in Section II.A) with the following L1-BDA algorithm. This algorithm was originally presented in [11] as a variant of principal component analysis.

1. Initialization: Pick any $\boldsymbol{w}(0) \neq 0$. Set $\boldsymbol{w}(0) \leftarrow \boldsymbol{w}(0)/||\boldsymbol{w}(0)||$ and $t = 0$.
2. Polarity check: For all i, if $\boldsymbol{w}^T(t)\boldsymbol{y}_i < 0$, $p_i(t) = -1$, otherwise $p_i(t) = 1$.
3. Flipping and maximization: Set $t \leftarrow t + 1$. $\boldsymbol{w}(t) = \sum_{i=1}^{n_y} p_i(t-1)\boldsymbol{y}_i$. Set $\boldsymbol{w}(t) \leftarrow \boldsymbol{w}(t)/||\boldsymbol{w}(t)||$.
4. Convergence check:

   a. If $\boldsymbol{w}(t) \neq \boldsymbol{w}(t-1)$, go to Step 2.

   b. Else if there exists i such that $\boldsymbol{w}^T(t)\boldsymbol{y}_i = 0$, set $\boldsymbol{w}(t) \leftarrow (\boldsymbol{w}(t) + \boldsymbol{dw})/||\boldsymbol{w}(t) + \boldsymbol{dw}||$ and go to Step 2. Here, $\boldsymbol{dw}$ is a small nonzero random vector.

   c. Otherwise, set $\boldsymbol{w}^{\star} = \boldsymbol{w}(t)$ and stop.

Theorem 1: With the above L1-BDA algorithm, the weight vector $\boldsymbol{w}(t)$ converges to $\boldsymbol{w}^{\star}$, which is a local maximum point of $\sum_{i=1}^{n_y} |\boldsymbol{w}^T \boldsymbol{y}_i|$.

Proof: Firstly, we can show that $\sum_{i=1}^{n_y} |\boldsymbol{w}^T(t)\boldsymbol{y}_i|$ is a non-decreasing function of t as the following:

$$\begin{aligned} \sum_{i=1}^{n_y} |\boldsymbol{w}^T(t)\boldsymbol{y}_i| &= \boldsymbol{w}^T(t)(\sum_{i=1}^{n_y} p_i(t)\boldsymbol{y}_i) \geq \boldsymbol{w}^T(t)(\sum_{i=1}^{n_y} p_i(t-1)\boldsymbol{y}_i) \\ &\geq \boldsymbol{w}^T(t-1)(\sum_{i=1}^{n_y} p_i(t-1)\boldsymbol{y}_i) = \sum_{i=1}^{n_y} |\boldsymbol{w}^T(t-1)\boldsymbol{y}_i|. \end{aligned} \tag{11}$$

In the above, the first inequality is due to the fact that $\{p_i(t)\}_{i=1}^{n_y}$ is the set of optimal polarity corresponding to $\boldsymbol{w}(t)$, such that for all i, $p_i(t)\boldsymbol{w}^T(t)\boldsymbol{y}_i \geq 0$. Note that the inner product of two vectors is maximized when the two vectors are parallel. Since $||\boldsymbol{w}(t)|| =$

$||\boldsymbol{w}(t-1)||$ $(=1)$ and the vectors $\boldsymbol{w}(t)$ $(=\frac{\sum_{i=1}^{n_y} p_i(t-1)\boldsymbol{y}_i}{||\sum_{i=1}^{n_y} p_i(t-1)\boldsymbol{y}_i||})$ and $\sum_{i=1}^{n_y} p_i(t-1)\boldsymbol{y}_i$ are parallel, the second inequality holds.

Because the objective function is non-decreasing and there are finite number of training samples, the L1-BDA algorithm converges to a weight $\boldsymbol{w}^\star$.

Secondly, we show that the objective function has a local maximum value at $\boldsymbol{w}^\star$. This can be shown as follows.

Because $\boldsymbol{w}(t)$ converges to $\boldsymbol{w}^\star$, $\boldsymbol{w}^\star p_i(t)\boldsymbol{y}_i \geq 0$ for all i. Since the number of training samples is finite and $\boldsymbol{w}^\star \boldsymbol{y}_i \neq 0$ for all i which is ensured by Step 4b, there exists a small neighborhood $N(\boldsymbol{w}^\star)$ of $\boldsymbol{w}^\star$ such that $\boldsymbol{w} p_i(t)\boldsymbol{y}_i \geq 0$ for all i and $\boldsymbol{w} \in N(\boldsymbol{w}^\star)$. Since $\boldsymbol{w}^\star$ is parallel to $\sum_{i=1}^{n_y} p_i(t)\boldsymbol{y}_i$, the inequality $\sum_{i=1}^{n_y} |\boldsymbol{w}^{\star T}\boldsymbol{y}_i| > \sum_{i=1}^{n_y} |\boldsymbol{w}^T\boldsymbol{y}_i|$ holds for all $\boldsymbol{w} \in N(\boldsymbol{w}^\star)$ and $\boldsymbol{w}^\star$ is a local maximum point.

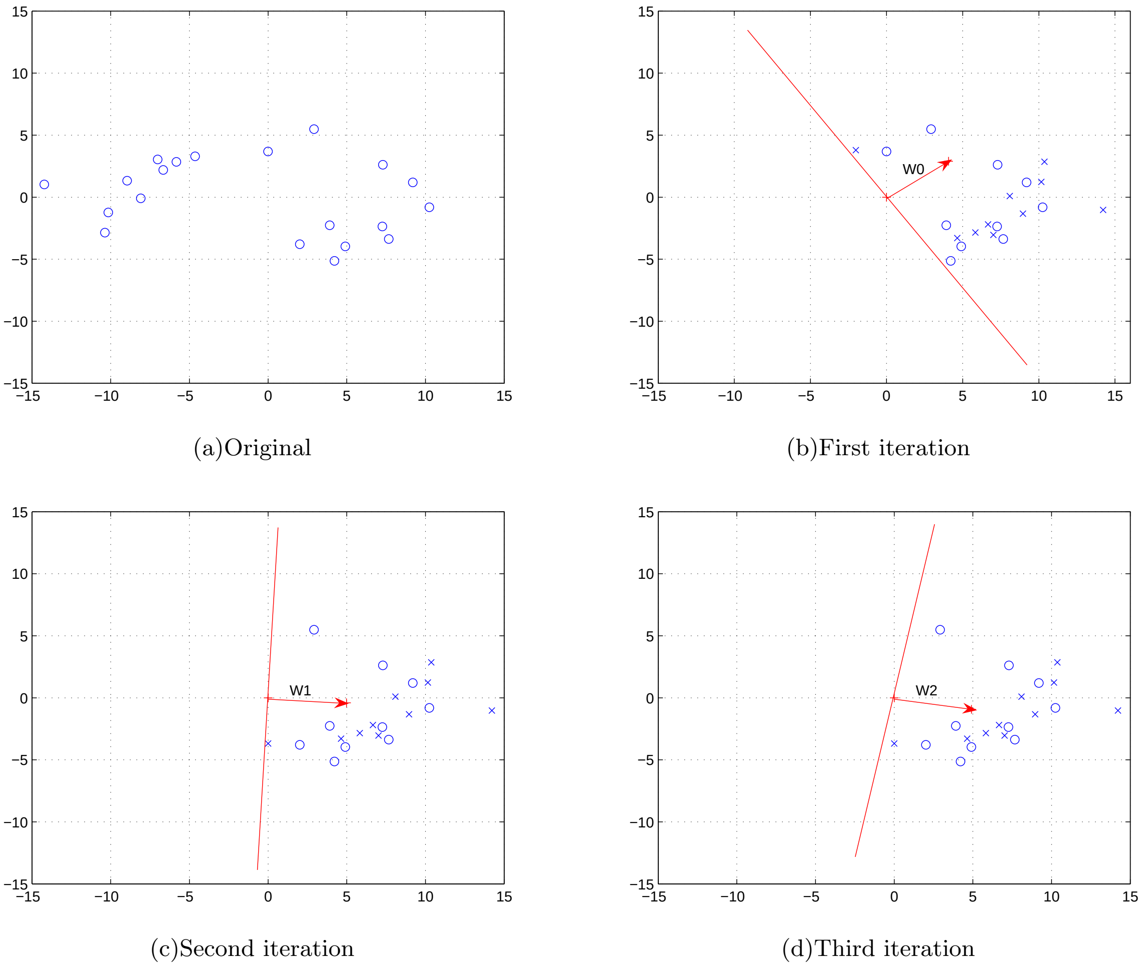
Therefore, the L1-BDA algorithm finds a local maximum point $\boldsymbol{w}^\star$. ■

Because the projection vector $\boldsymbol{w}(t)$ is determined by a linear combination of data points $\boldsymbol{y}_i$'s, i.e., $\boldsymbol{w}(t) \propto \sum_{i=1}^{n} p_i(t-1)\boldsymbol{y}_i$, the proposed algorithm is naturally invariant to rotations.

Note that this algorithm tries to find a local maximum solution and there is a possibility that it may not be the global solution. Note also that the initial value $\boldsymbol{w}(0)$ can be set arbitrarily. By setting the initial value appropriately, e.g., $\boldsymbol{w}(0) = \text{argmax}_{\boldsymbol{y}_i} ||\boldsymbol{y}_i||$ or the solutions of the conventional BDA (L2 solution) or SBDA, we expect to find the global maximum point with higher probability in fewer iterations.

The L1-BDA algorithm is described by an example in Fig. 3. In this example, we assume that the dataset was preprocessed by the sphering process and we focus only on the negative samples. Fig. 3 (a) is the negative samples of the original dataset which has been created as follows. Firstly, 20 random samples $\{(a_i, b_i)\}_{i=1}^{20}$ were generated in a two dimensional space with the mean distance of 5 from the origin and the variance of 1 with a uniform random orientation. And the point (a_i, b_i) is transformed to $(2a_i, b_i)$ for all i.

If we set the initial weight $\boldsymbol{w}(0) = [0.8151, 0.5794]^T$ as shown in Fig. 3 (b), the polarities of the points located below the line are set to -1 in the polarity checking step and these are flipped across the origin and marked as 'x'. By averaging all the points marked as 'o' and 'x', we get a new weight $\boldsymbol{w}(1) = [0.9967, -0.0812]^T$ as shown in Fig. 3 (c). By the same procedure, we get $\boldsymbol{w}(2) = [0.9826, -0.1859]^T$ as shown in Fig. 3 (d). After this, the



(a)Original

(b)First iteration

(c)Second iteration

(d)Third iteration

Fig. 3. L1-BDA procedure

polarity of each point does not change any more and the convergence condition is fulfilled. Thus, $\boldsymbol{w}(2)$ becomes $\boldsymbol{w}^{\star}$, which is the global maximum point in this case.

We initialized the sample points as well as the weight $\boldsymbol{w}(0)$ randomly for this example and the solution was found in 3.26 iterations on average with the standard deviation of 0.8 for 1,000 experiments. On the other hand, it required only 3 iterations in all 1000 experiments when we set the initial weight as $\boldsymbol{w}(0) = \text{argmax}_{\boldsymbol{y}_i} ||\boldsymbol{y}_i||$.

For this L1-BDA algorithm, we can also apply the saturation scheme in the previous subsection in order to further suppress the effect of the too far-away negative samples as follows:

1. Initialization: Pick any $\boldsymbol{w}(0) \neq 0$. Set $\boldsymbol{w}(0) \leftarrow \boldsymbol{w}(0)/||\boldsymbol{w}(0)||$ and $t = 0$.
2. Polarity check and saturation: For all i, if $\boldsymbol{w}^T(t)\boldsymbol{y}_i < 0$, $p_i(t) = -1$, else $p_i(t) = 1$.

$$\boldsymbol{z}_i = \begin{cases} \boldsymbol{y}_i, & \text{if} \quad \boldsymbol{w}^T(t)\boldsymbol{y}_i \leq \gamma \\ \frac{\gamma}{|\boldsymbol{w}^T(t)\boldsymbol{y}_i|}\boldsymbol{y}_i, & \text{if} \quad \boldsymbol{w}^T(t)\boldsymbol{y}_i > \gamma. \end{cases}$$

3. Flipping and maximization: Set $t \leftarrow t+1$. $\boldsymbol{w}(t) = \sum_{i=1}^{n_y} p_i(t-1)\boldsymbol{z}_i$. Set $\boldsymbol{w}(t) \leftarrow \boldsymbol{w}(t)/||\boldsymbol{w}(t)||$.

4. Convergence check:

a. If $||\boldsymbol{w}(t) - \boldsymbol{w}(t-1)|| \geq \tau$, go to Step 2.

b. Else if there exists i such that $\boldsymbol{w}^T(t)\boldsymbol{y}_i = 0$, set $\boldsymbol{w}(t) \leftarrow (\boldsymbol{w}(t) + \boldsymbol{dw})/||\boldsymbol{w}(t) + \boldsymbol{dw}||$ and go to Step 2. Here, $\boldsymbol{dw}$ is a small nonzero random vector.

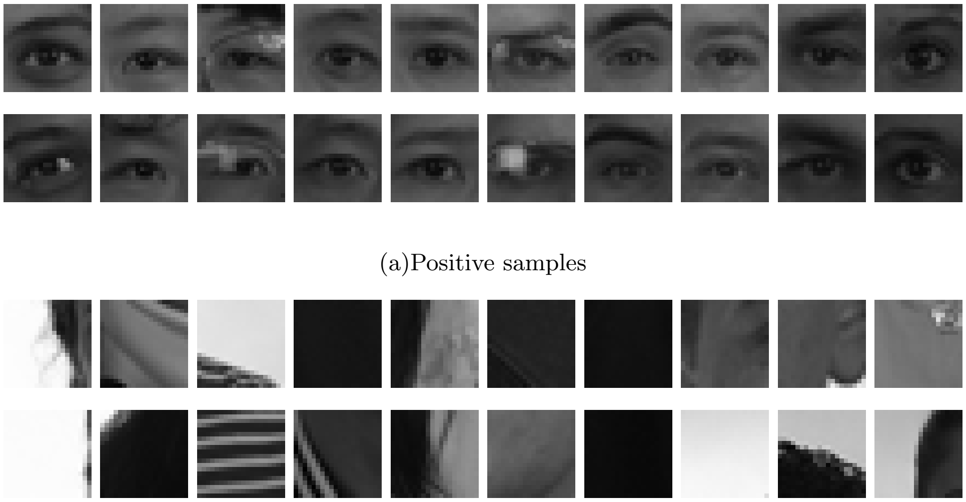
c. Otherwise, set $\boldsymbol{w}^\star = \boldsymbol{w}(t)$ and stop.

In this saturated version of L1-BDA algorithm (SL1-BDA), the saturation scheme is applied in every iterations in Step 2. By doing this, the error in the solution can be reduced.

Note that the convergence criterion in Step 4 is different from the convergence criterion in the original L1-BDA algorithm. This is because the convergence speed decreases compared to the original L1-BDA algorithm when we use the saturation scheme in every iterations. However, by incorporating an appropriate tolerance τ, the convergence speed will not be a serious problem in the saturated L1-BDA algorithm. As an alternative, instead of setting a tolerance τ, we can set a maximum number of iterations t_{max} to an appropriate value.

Until now, we have shown that we can extract one best feature that maximizes the L1 objective function. The proposed method can be easily extended to extract arbitrary number of features by applying the same procedure to the remainder of the projected samples as follows:

```
$\boldsymbol{w}_0 = \boldsymbol{0}$, $\{\boldsymbol{y}_i^0 = \boldsymbol{y}_i\}_{i=1}^{n_y}$.
For $m = 1$ to $M$,
    For all $i$, $i = 1 \cdots n_y$, $\boldsymbol{y}_i^m = \boldsymbol{y}_i^{m-1} - (\boldsymbol{w}_{m-1}^T \boldsymbol{y}_i)\boldsymbol{w}_{m-1}$.
    In order to find $\boldsymbol{w}_m$, apply the L1-BDA or saturated L1-BDA algorithms to $\{\boldsymbol{y}_i^m\}_{i=1}^{n_y}$.
end
```



(a)Positive samples

(b)Negative samples

Fig. 4. Eye and noneye samples for training

IV. Experimental Result

In order to show the effectiveness of the proposed methods, we have applied the proposed feature extraction methods to various real world problems and compared the performances with other conventional feature extraction algorithms.

A. Eye dataset in FERET database

This dataset was generated from the Color FERET face database [12] to distinguish whether an image is an eye or not. In each face image in the FERET database, both left and right eyes were selected for positive samples. On the other hand, negative samples were randomly chosen from the other regions of images. The training set consists of 400 positive and 400 negative samples obtained from the fa images of the FERET database while the test set contains 400 positive and 400 negative samples from the fb images of FERET database. Figure 4 shows some positive and negative samples in the training set. Each positive sample was cropped in proportion to the distance between two eyes and was rescaled to a size of 20×20. The negative samples were also cropped with the same size. Each pixel was considered as an input variable constituting a 400-dimensional input space.

For this dataset, we compared the classification performances of the proposed methods,

TABLE I

Classification rates for FERET Eye dataset. (1-NN)

# of features	LDA	Cher. LDA	BDA	SBDA ($\gamma = 1$)	L1BDA	SL1BDA ($\gamma = 1$)
1	**86.875**	**84.625**	81.875	86.25	87.625	86.625
2	–	83.75	88	95	91.125	94.5
3	–	81.375	91.25	96.625	96.375	96.75
4	–	78.75	93	97.5	97.125	97.625
5	–	80.625	93	98.25	97.375	98.25
6	–	79.75	94.875	**98.625**	97.375	**98.625**
7	–	78	94.75	98.5	97.875	98.125
8	–	79.125	95	98.125	**98**	98.125
9	–	78.625	95	98.375	98	98.5
10	–	79.75	**95.875**	98	97.875	98.125

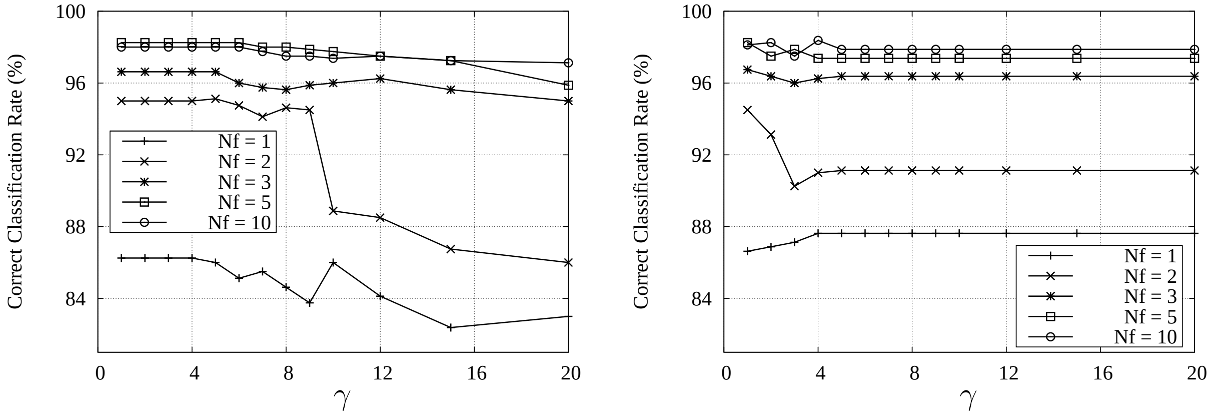
SBDA, L1-BDA, and SL1-BDA with those of LDA, BDA and Chernoff LDA [3] in Table I.

Because this is a binary classification problem, LDA can extract only one feature while the other methods can produce more features. For the other methods, we extracted up to 10 features. For the Chernoff LDA and BDA family, the regularization method of (5) was used with $\alpha = 0.1$. As a classifier, the one nearest neighbor (1-NN) classifier was used. For SBDA and SL1-BDA, the performance depends on the value of γ. In the table for both SBDA and SL1-BDA, $\gamma = 1$ was used. The best classification rates of each method were shown in bold faces.

In this problem, the positive samples look very similar, while the negative samples show little resemblance. Therefore, this problem fits better to the BDA family than the conventional LDA and its variants. As a result, we can see that the best classification rates of BDA, SBDA, L1-BDA, and SL1-BDA are higher than those of LDA and Chernoff LDA by around 10%. However, when the number of extracted feature is one, the BDA family does not show noticeable performance difference from LDA. The reason for this can be ascribed to the use of the 1-NN classifier. Because LDA tries to congregate negative samples as well as positive ones, the projection of the samples are separated in two distinct regions: positive and negative. On the other hand, in BDA family, positive samples are highly concentrated near the center, whereas negative samples will be randomly distributed apart from the center like clouds. Although most of the negative samples will be far away from the center, some will be close to the center, especially when the number of feature is very small, and 1-NN classifier will perform poorly in this case. Therefore, it is more likely that 1-NN will perform worse for the BDA family than for the LDA. Actually, when we applied 3-NN classifier, the correct classification rates for BDA, SBDA, L1-BDA, and SL1-BDA using only one feature were raised to 85.375%, 89.375%, 88.5%, and 89.375% respectively, while they were 87.5% and 86.125% for LDA and Chernoff LDA respectively.

Comparing the performances of the proposed methods (SBDA, L1-BDA, and SL1-BDA) with that of BDA, we can see that SBDA, L1-BDA, and SL1-BDA outperformed the conventional BDA regardless of the number of extracted features. This shows that the negative samples far away from the center of the positive samples have a detrimental effect on the performance of BDA and this problem is alleviated by the proposed methods. Comparing the three proposed methods, we can see that SBDA and SL1-BDA performed slightly better than L1-BDA for this problem except when the number of extracted features was one. Comparing SBDA and SL1-BDA, the performance difference was negligible. From this result, we can infer that once the saturation scheme was applied, whether optimization was performed in L2-sense (SBDA) or in L1-sense (SL1-BDA), the classification performance were almost the same for this problem because all the samples were already moved near the center of the positive samples by the saturation scheme and there is no single sample that affects the optimization function dominantly. However, without the saturation scheme, L1-optimization (L1-BDA) was more effective than L2-optimization (BDA) in reducing the effect of the negative samples far away from the positive mean.

To see the effect of γ in SBDA and SL1-BDA, various values of γ from 1 to 20 was tested with various numbers of extracted features (1, 2, 3, 5, and 10) and the performances were shown in Figure 5. In the figure, we can see that as γ increases the performance of SBDA decreases slightly but gradually when $\gamma > 4$ for SBDA. However, the performance of SL1-BDA does not decrease significantly for large values of γ. When we checked the norms of negative samples in the training data, all the 400 negative samples had their norms larger than 4 and the numbers of samples whose norm exceeds 5, 10, and 20 were 396,



(a)SBDA

(b)SL1-BDA

Fig. 5. The effect of γ for SBDA and SL1-BDA

344, and 234 respectively. This shows the reason for a constant performance of SBDA when $\gamma \leq 4$. However, for SL1-BDA, the saturation scheme is applied in each iteration with different initial weight and the performances were different for $\gamma \leq 4$. Comparing Fig. 5 (b) and Table I, we can see that the performances of SL1-BDA were the same as those of L1-BDA for $\gamma \geq 5$. This shows that negative samples with large norm does not affect the performance of L1-BDA heavily.

B. ETH-80 dataset

Object categorization is a classical problem in computer vision. In this section, we applied the proposed feature extraction algorithms to the ETH-80 database [13] which is for object categorization. It contains images of the eight categories: apple, car, cow, cup, dog, horse, pear, and tomato. For each category, there are ten objects, each of which contains 41 images from different viewpoints. Overall, the database contains 3,280 images of 80 objects. The examples of the 80 objects are shown in Fig. 6.

We first generated subimages corresponding to the smallest rectangle that includes the boundary of the object by cropping the original images. Since each subimage is different in size, we rescaled each of them into a size of 30 × 25 pixels. Each pixel was considered as an input variable and therefore, each image corresponds to a point in a 750-dimensional input space. Each of 8 categories was regarded as a class. Thus, it became an eight-class classification problem.

Because the BDA family was originally developed for one-class problems, we should

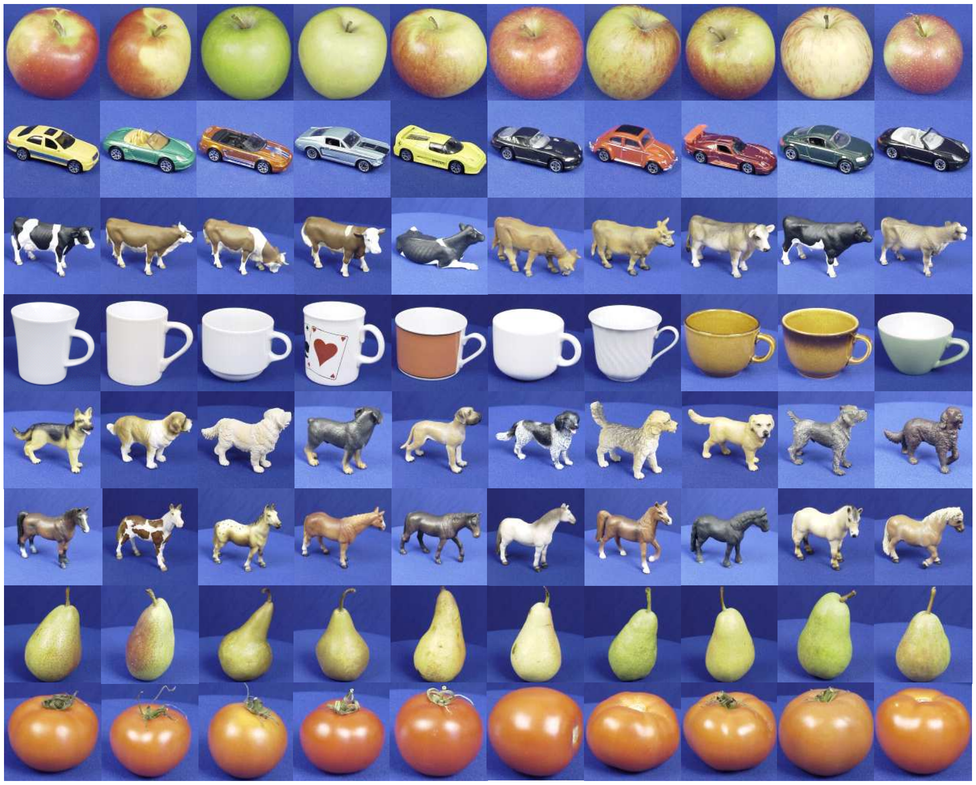


Fig. 6. 80 objects of ETH-80 databases

find ways to extend the BDA family to multiclass problems in order to apply them to this ETH-80 problem. Many researches have been conducted to extend binary classifiers to multi-class classification problems [14] [15] [16] [17], but the simplest way of extending the BDA family to D-class classification problems is to construct D datasets with only two classes (positive and negative) as shown in Fig. 7. In constructing the i-th dataset, the samples from the i-th class are regarded as the positive samples and the rest are regarded as the negative samples. Then we apply the BDA family to each of these datasets as shown in Fig. 7.

Note that in Fig. 7, the i-th dataset focuses only on the i-th class and the samples from the other classes are regarded as negative ones. As a result, if M_i features are extracted for the i-th dataset, the total number of extracted features becomes $\mathcal{M} = \sum_{i=1}^{D} M_i$. Note that the features for the i-th dataset may not be orthogonal to features of the other datasets. Also, $\mathcal{M}$ may be larger than the dimension of the original input space d.

Once the features are extracted, the classification can be done by various classifiers such

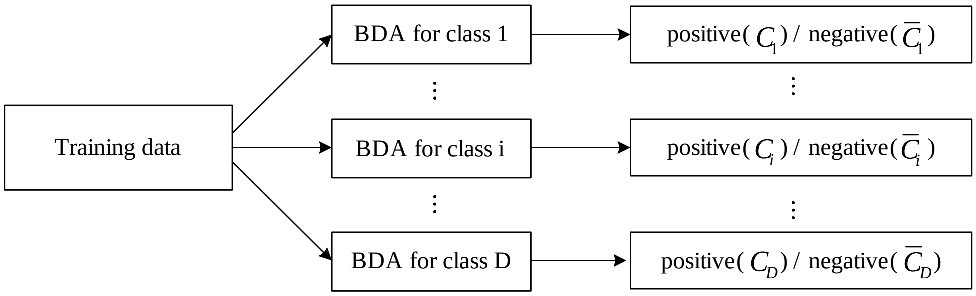

Training data
BDA for class 1
BDA for class i
BDA for class D
positive(C_1) / negative($\bar{C}_1$)
positive(C_i) / negative($\bar{C}_i$)
positive(C_D) / negative($\bar{C}_D$)

Fig. 7. BDA for D-class classification problem

as multi-layer perceptrons, decision trees, nearest neighborhood (NN) and so on.

We used this extension of the BDA family in order to classify 8 categories of the ETH-80 database. For the test, we performed the leave-one-object-out test. In other words, 79 objects were used in the training and the remaining one object was used as a test set. In this way, 80 tests were performed in total, each of which contains 3,239 training samples and 41 test samples.

Before applying feature extraction algorithms, in order to reduce the processing time, we preprocessed the data with PCA and obtained the best 60 features corresponding to the highest eigenvalues. On this 60 preliminary features we have applied various feature extraction algorithms such as LDA, Chernoff LDA, BDA, SBDA, L1-BDA, and SL1-BDA.

Figure 8 shows the classification performances of various feature extraction methods. The numbers of extracted features were also varied from 5 to 60 for each method. For the BDA family, the numbers of extracted features shown in the figure are the numbers for each class; therefore, the total numbers of features are 8 times the numbers shown in the figure. As a classifier, the 1-NN classifier was used. For SBDA and SL1-BDA, the parameter γ was set to 1.

Because it is an eight-class classification problem, LDA can produce only 7 features. The best performance of LDA was 65.18% when the number of extracted features was 7. Due to the difference in the number of extracted features and its low performances, the classification results of LDA were omitted in the figure.

In the figure, we can see that SBDA performed best regardless of the number of features and its best classification rate reached 74.05%. L1-BDA and SL1-BDA were also better than BDA regardless of the number of extracted features and their best classification rates

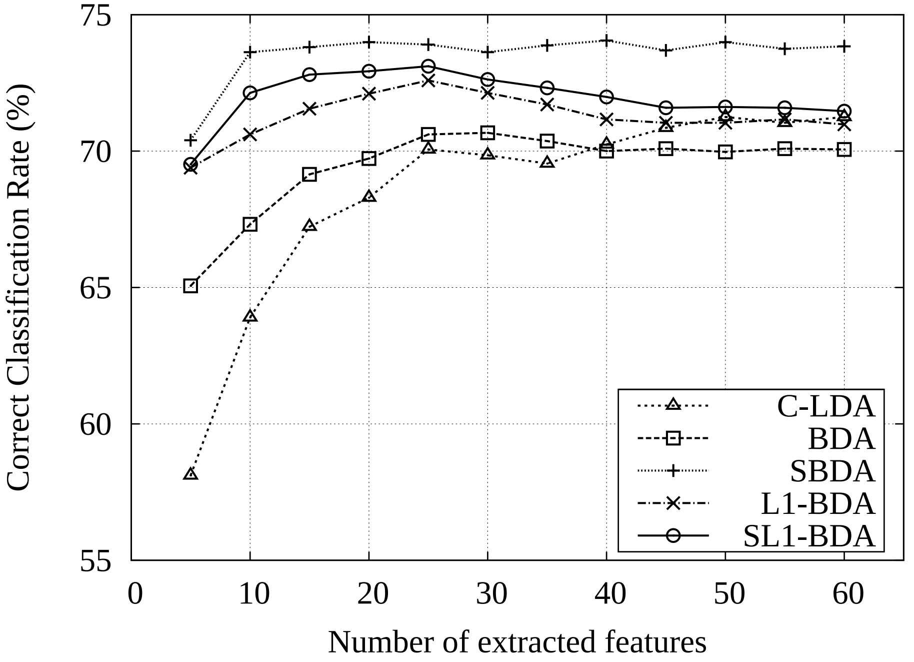


Fig. 8. Correct classification rates for ETH-80 dataset (using 1-NN classifier).

were 72.59% and 73.11% respectively. The best classification rates of Chernoff-LDA was 71.25%. Regardless of the number of extracted features, SL1-BDA was better than L1BDA but it was worse than SBDA. From this result, we can see that the saturation scheme effectively reduces the bad effect of the negative samples with large norm. By optimizing L1-norm instead of L2-norm, we could also get better classification performance. However, we can see that combining saturation scheme with L1-norm optimization was not always better than applying the saturation scheme alone. Instead, once the saturation scheme was applied, optimizing L2-norm was better than optimizing L1-norm for this problem.

C. UCI datasets

We also applied SBDA, L1-BDA, and SL1-BDA to several datasets in UCI machine learning repositories [18] and compared their performances with those of other conventional feature extraction methods such as BDA, LDA and Chernoff LDA.

Table II shows a brief summary of the datasets used in this paper. These datasets have been used in many studies [3] [19] [20]. As a classifier, 1-NN classifier was used.

For each dataset except the 'Glass', we performed 10-fold cross validation (CV) 10

TABLE II

UCI Data sets used in the experiments

Data set	# of variables	# of class	# of instances
Austrailian	14	2	690
Balance	4	3	625
Breast cancer	9	2	683
Glass	9	6	214
Heart disease	13	2	297
Iris	4	3	150
Liver	6	2	345
Pima	8	2	768
Sonar	60	2	208
Vehicle	18	4	846

times and computed the average classification rate and its standard deviation. Because the ‘Glass’ dataset has relatively small number of samples (i.e., 214) compared to its six classes and its data distribution is highly unbalanced such that there are only 9 samples for the third class whereas the fourth class includes 76 samples, we used 3-fold CV instead of 10-fold CV. Before training, each input variable in the training set was normalized to have zero mean and unit variance. Of course, the variables in the test set were also normalized using the means and the variances of the training set. For the SBDA and SL1-BDA γ was set to 1. For each feature extraction method, the regularization method of (5) was used with the regularization parameter $\alpha = 0.1$. Because the datasets are not one-class problems, for the BDA family, the extension method shown in Fig. 7 was used and the dimensions of different subspaces were set to be the same, i.e., $M_i = M$, for $i = 1, \cdots, D$ where D is the number of classes.

Table III shows the performance of various feature extraction methods. The performances shown in the table are the best classification rates among the results obtained by varying the numbers of extracted features from one to $d-1$ (for LDA) or d (for the other methods). Here, d is the number of original input variables. The numbers in the parentheses are the optimal numbers of features. For LDA and Chernoff LDA, these numbers represent the dimension of the feature vector, whereas for BDA family, these are the dimensions of one subspace, M, obtained by applying the corresponding BDA algorithm to

TABLE III

Experimental Results on UCI datasets (1-NN classifier)

Data set	LDA	Chernoff LDA	BDA	SBDA ($\gamma = 1$)	L1BDA	SL1BDA ($\gamma = 1$)
Austrailian	80.55±1.02 (1)	81.35±0.82 (6)	80.68±1.28 (3)	81.29±0.74 (4)	81.57±1.19 (4)	**81.67±1.19 (3)**
Balance	88.00±0.49 (2)	87.68±0.79 (2)	91.25±0.99 (3)	95.73±0.62 (2)	**95.76±0.64 (2)**	94.38±0.87 (3)
Breast	96.03±0.34 (1)	**96.34±0.35 (2)**	95.94±0.44 (2)	96.28±0.29 (6)	96.05±0.62 (2)	96.09±0.20 (6)
Glass	62.85±1.73 (5)	63.55±1.43 (9)	66.19±2.21 (3)	**70.65±1.12 (6)**	67.66±2.32 (3)	68.74±1.37 (8)
Heart	76.43±1.19 (1)	76.60±1.78 (1)	76.73±1.41 (2)	76.73±1.41 (2)	**77.44±1.41 (4)**	**77.44±1.41 (4)**
Iris	96.93±0.72 (1)	97.13±0.77 (1)	**97.20±0.69 (1)**	96.87±0.63 (1)	97.13±0.77 (1)	96.60±0.21 (4)
Liver	61.01±3.28 (1)	**65.65±2.50 (4)**	65.04±2.20 (3)	65.42±2.24 (3)	65.13±1.39 (5)	65.13±1.39 (5)
Pima	69.13±1.32 (1)	69.78±0.51 (8)	69.79±0.74 (5)	70.01±0.56 (5)	**70.42±1.55 (4)**	**70.42±1.55 (4)**
Sonar	73.03±2.27 (1)	81.06±2.25 (54)	78.56±2.03 (5)	84.86±1.59 (18)	80.86±1.28 (5)	**84.90±1.51(17)**
Vehicle	74.33±1.10 (3)	**81.55±0.51 (9)**	73.45±0.66 (5)	76.32±0.75 (15)	74.65±0.46 (2)	80.01±0.50 (3)
Average	77.83	80.07	79.48	81.41	80.67	**81.54**

one of the binary problems shown in Fig. 7. Therefore, the total number of the extracted features for BDA family was $\mathcal{M} = M \times D$. The best classification rate for each dataset was indicated in boldface. At the last row, the average of 10 datasets was reported for each method.

In the table, we can see that the performances of SBDA, L1-BDA, and SL1-BDA were better compared to the conventional BDA for all the datasets except for Iris. Even in this case, we can see that the performance differences were very small. The performance of the proposed methods were better than that of BDA at least 1% on average. The performance of BDA was better than that of LDA on average but it was slightly worse than that of Chernoff LDA. Comparing the performances of the proposed methods, SBDA and SL1-BDA were slightly better than L1-BDA on average and the two showed almost the same results.

Comparing the proposed algorithms with LDA, we can see that the proposed algorithms are better than LDA for all the datasets except for Iris. For Iris dataset, all the methods showed almost the same classification rate. Although the performance enhancements are rather small, the performances of SBDA, L1-BDA, and SL1-BDA were slightly better than that of Chernoff LDA on average.

V. Conclusions

In this paper, we proposed two methods to enhance the performance of BDA which was originally proposed for one-class problems.

The first method, SBDA tries to reduce the negative effect in extracting features due to the negative samples which are very far away from the center of positive samples. By restricting the distance between a negative sample and the center of positive samples to a moderate value, SBDA can produce good features for better classification performance.

The second method, L1-BDA tries to solve the same problem by using the objective function based on the L1 norm instead of the L2 norm. In due course, a new method of L1-norm optimization was introduced and was proven to find a local maximum point. The proposed L1-norm optimization technique is intuitive, simple, and easy to implement.

These two methods successfully suppress the negative effect of the too far away negative samples from the center of positive samples without adding much complexity. The two schemes were combined to produce a third method, SL1-BDA and a simple method that extends the applicability of the BDA family to multi-class classification problems was also provided.

We have applied the proposed methods to several classification problems and compared the performance with the conventional methods. By applying the proposed methods, we could obtain better classification performances than the conventional BDA, LDA and Chernoff LDA on average.

Acknowledgments

This work was supported by the Korea Research Foundation Grant funded by Korean Government (KRF-2007-313-D00610).

References

[1] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, second edition, 1990.

[2] T. Okada and S. Tomita, “An optimal orthonormal system for discriminant analysis,” *Pattern Recognition*, vol. 18, no. 2, pp. 139–144, Feb. 1985.

[3] Marco Loog and Robert P.W. Duin, “Linear dimensionality reduction via a heteroscedastic extension of lda: The chernoff criterion,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 732–739, June 2004.

[4] M. Zhu and A.M. Martinez, "Subclass discriminant analysis," *IEEE Trnasactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1274–1286, Aug. 2006.

[5] Xiang Sean Zhou and Thomas S. Huang, "Small sample learning during multimedia retrieval using biasmap," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, Hawaii, USA, December 2001, vol. 1, pp. 11–17.

[6] Juwei Lu, K.N. Plataniotis, and A.N. Venetsanopoulos, "Regularization studies of linear discriminant analysis in small sample size scenarios with applications to face recognition," *Pattern Recognition Letters*, vol. 26, pp. 181–191, 2005.

[7] L.F. Chen, H.Y.M. Liao, M.T. Ko, J.C. Lin, and G.Y. Yu, "A new lda-based face recognition system which can solve the small sample size problem," *Pattern Recognition*, vol. 33, pp. 1713–1726, 2000.

[8] J. Yang, F. Frangi, J.-Y. Yang, D. Zhang, and Z. Jin, "Kpca plus lda: A complete kernel fisher discriminant framework for feature extraction and recognition," *IEEE Trnasactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 230–244, Feb. 2005.

[9] J.H. Friedman, "Regularized discriminant analysis," *Journal of American Statistics Association*, vol. 84, pp. 165–175, 1989.

[10] Q. Ke and T. Kanade, "Robust subspace computation using l1 norm," Aug. 2003, http://citeseer.ist.psu.edu/ke03robust.html.

[11] Nojun Kwak, "Principal component analysis based on l1 norm maximization," *submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.

[12] P.J. Phillips, H. Moon, S.A. Rizvi, and P.J. Rauss, "The feret evaluation methodology for face recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, , no. 10, pp. 1090–1104, October 2000.

[13] Bastian Leibe and Bernt Schiele, "Analyzing appearance and contour based methods for object categorization," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'03)*, Madison, WI, June 2003.

[14] Chih-Wei Hsu and Chih-Jen Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, March 2002.

[15] Ting-Fan Wu, Chie-Jen Lin, and Ruby C. Weng, "Probability estimation for multi-class classification by pairwise coupling," *Journal of Machine Learning Research*, vol. 4, pp. 975–1005, August 2004.

[16] David M.J. Tax and Robert P.W. Duin, "Using two-class classifiers for multiclass classification," in *Proc. 16th International Conference on Pattern Recognition*, Quebec City, Canada, August 2002, vol. 2, pp. 124–127.

[17] William Smart and Mengjie Zhang, "Using genetic programming for multiclass classification by simultaneously solving component binary classification problems," *Lecture Notes in Computer Science*, vol. 3447, pp. 227–239, 2005.

[18] D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz, "Uci repository of machine learning databases," 1998, http://www.ics.uci.edu/ mlearn/MLRepository.html.

[19] H. Xiong, M.N.S. Swamy, and M.O. Ahmad, "Optimizing the kernel in the empirical feature space," *IEEE Transactions on Neural Networks*, vol. 16, no. 2, pp. 460–474, March 2005.

[20] C.J. Veeman and M.J.T. Reinders, "The nearest subclass classifier: A compromise between the nearest mean and nearest neighbor classifier," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 9, pp. 1417–1429, Sep. 2005.