Re-weighting Linear Discrimination Analysis under Ranking Loss

Yong Ma, Yoshihisa Ijiri, Shihong Lao, Masato Kawade
Core Technology Center, Omron Corporation
Kyoto, 619-0283, Japan
{ma,joyport,lao,kawade}@ari.ncl.omron.co.jp

Abstract

Linear Discrimination Analysis (LDA) is one of the most popular feature extraction and classifier design techniques. It maximizes the Fisher-ratio between between-class scatter matrix and within-class scatter matrix under a linear transformation, and the transformation is composed of the generalized eigenvectors of them. However, Fisher criterion itself can not decide the optimum norm of transformation vectors for classification. In this paper, we show that actually the norm of the transformation vectors has strong influence on classification performance, and we propose a novel method to estimate the optimum norm of LDA under the ranking loss, re-weighting LDA. On artificial data and real databases, the experiments demonstrate the proposed method can effectively improve the performance of LDA classifiers. And the algorithm can also be applied to other LDA variants such as Non-parametric Discriminant Analysis (NDA) to improve theirs performance further.

1. Introduction

Linear Discriminant Analysis (LDA) is a well-known method for dimensionality reduction and classification that projects high-dimensional data onto a low dimensional space where the data achieves maximum class separability [1, 2, 3]. The optimal projection or transformation in classical LDA is obtained by minimizing the within-class scatter matrix and maximizing the between-class scatter matrix simultaneously (Fisher criterion), which is equivalent to solving a generalized eigen-decomposition problem; and the optimal projection consists of the eigenvectors (hereafter LDA vectors for short) corresponding to the first several largest eigenvalues. Even in the presence of more advanced and sophisticated classification techniques, LDA has not left the minds of researchers. It has been applied successfully in many applications including face recognition [4], document classification [5], and microarray gene expression analysis [6]. And many researches are still being conducted to improve its performance further.

The proposed LDA-related researches mainly focus on several drawbacks of classical LDA. One is the singularity of scatter matrix due to the small sample size [6, 8]. The small sample size arises whenever the number of samples is smaller than the feature dimensionality, which causes within-class or between-class scatter matrix to be singular, and makes it difficult to get stable eigenvectors. Several extensions have been proposed to overcome this problem. These include orthogonal LDA (OLDA) [6, 19], uncorrelated LDA [6, 19], null space LDA [10], regularized LDA [9], penalized LDA [21], etc. The small sample size problem can also be overcome by applying the PCA before LDA [4] at the cost of some discriminant information.

Another limitation arises from the assumption of the LDA that all classes should have a Gaussian distribution with a single shared covariance. LDA guaranteed to find the best directions when this assumption is valid. Or else, if the class distributions are multimodal and share the same mean, it will fail to find the discriminant direction [2]. To overcome this limitation, some extensions also have been proposed, such as nonparametric discriminant analysis (NDA) [10], stepwise nearest neighbor discriminant analysis [8], and heteroscedastic LDA [7].

However, to the best of our knowledge, very little research has been conducted on the relation between norms of LDA vectors and classification performance. We know that in LDA framework, there exists numerous eigenvectors corresponding to one eigenvalue, and all these eigenvectors are optimal with regard to Fisher criterion [2]. In other words, the norms of LDA vectors do not contribute to the Fisher criterion. Two questions arise naturally: Are the norms of LDA vectors irrelevant to the classification performance? How to decide the optimum norms if they are really relevant?

We will answer the above questions in this paper. First we will demonstrate that actually the norms of LDA transformation vectors have strong influence on the classification performance. Inspired by this fact, we design a special loss function based on sample-class pair ranking to learn the optimum norms of the LDA vectors. This loss function is a tradeoff between better generalization ability and minimum empirical ranking error of sample-class pairs. So after deciding the optimum norms of LDA vectors, the

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

newly learned transformation is not only optimal with regard to Fisher criterion, but also optimal in the sense of loss function. Because the proposed algorithm is implemented after the LDA decomposition, we call it Reweighting LDA (RW-LDA in short).

The contribution of this paper includes the following points. Firstly, in section 2 we reveal the fact ignored by many researchers, that norm of transformation vectors are important to the classification performance; secondly, we propose the novel re-weighting LDA algorithm to obtain the optimum norm under the ranking loss in section 3; thirdly, we thoroughly compare the proposed algorithm with other methods in the domains of metric learning and ranking learning in section 4; lastly, we report the experimental results using a collection of face images to evaluate the proposed method in section 5. Experimental results demonstrate the effectiveness of the proposed RW-LDA algorithm to improve the performance of not only classical LDA, but also for NDA and other LDA variations.

2. Review and Analysis of LDA

Considering a multi-class classification problem, a dataset is given that consists of n samples and c classes, $\{(\mathbf{x}_{11},\cdots,\mathbf{x}_{1n_1}),\ (\cdots),(\mathbf{x}_{c1},\cdots,\mathbf{x}_{cn_c})\}$, where $\mathbf{x}_{ij}\in\mathbb{R}^{\mathrm{H}}$ denotes the j-th ($1\le j\le n_i$) sample of i-th ($1\le i\le c$) class, n_i is the sample size of the i-th class, $n=\sum_{i=1}^{c}n_i$, and H is the data dimensionality. The between-class scatter matrix S_B and within-class scatter matrix S_W are defined as follows:

$$\mathbf{S}_B=\frac{1}{n}\sum_{i=1}^{c}n_i\left(\mathbf{m}_i-\mathbf{m}\right)\left(\mathbf{m}_i-\mathbf{m}\right)^T \tag{1}$$

$$\mathbf{S}_W=\frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n_i}\left(\mathbf{x}_{ij}-\mathbf{m}_i\right)\left(\mathbf{x}_{ij}-\mathbf{m}_i\right)^T \tag{2}$$

where $\mathbf{m}_i$ is the centroid of the i-th class, and $\mathbf{m}$ is the global mean of the whole dataset.

Classical LDA computes a linear transformation $\mathbf{P}=[\mathbf{p}_1,\cdots,\mathbf{p}_h]\in\mathbb{R}^{H\times h}$ that maps $\mathbf{x}_{ij}$ to $\mathbf{x}_{ij}^*$ in a h-dimensional space: $\mathbf{x}_{ij}^*=\mathbf{P}^T\mathbf{x}_{ij}$, and maximizes the class separability in this lower-dimensional space. In this space, the scatter matrices are

$$\mathbf{S}_B^*=\mathbf{P}^T\mathbf{S}_B\mathbf{P} \tag{3}$$

$$\mathbf{S}_W^*=\mathbf{P}^T\mathbf{S}_W\mathbf{P} \tag{4}$$

Generally, $trace(\mathbf{S}_W^*)$ or $\left|\mathbf{S}_W^*\right|$ measure the within-class cohesion, while $trace(\mathbf{S}_B^*)$ or $\left|\mathbf{S}_B^*\right|$ measures the between-class separation. Two favorable optimization criteria, also known as Fisher criterion [1, 2], are:

$$\arg\max_{\mathbf{P}}\frac{\left|\mathbf{P}^T\mathbf{S}_B\mathbf{P}\right|}{\left|\mathbf{P}^T\mathbf{S}_W\mathbf{P}\right|} \tag{5}$$

or

$$\arg\max_{\mathbf{P}}\{trace((\mathbf{P}^T\mathbf{S}_W\mathbf{P})^{-1}\mathbf{P}^T\mathbf{S}_B\mathbf{P})\}. \tag{6}$$

Provided that the with-in class scatter matrix $\mathbf{S}_W$ is nonsingular, these two criteria lead to the same optimal $\mathbf{P}$ which consists of the h eigenvectors of $\mathbf{S}_W^{-1}\mathbf{S}_B$ corresponding to its h largest eigenvalues [1, 2]. And

$$\mathbf{S}_B\mathbf{p}_i=\lambda_i\mathbf{S}_W\mathbf{p}_i \tag{7}$$

In the LDA-transformed space, nearest mean classifier (NMC) or nearest neighbor classifier (NNC) are often used to classify a sample $\mathbf{x}$ according to some metrics, such as Euclidean distance, normalized correlation, etc. For simplicity and efficiency, we only discuss the NMC using Euclidean distance after LDA projection in this paper. The decision function of the whole framework to classify $\mathbf{x}$ can be summarized as:

$$\operatorname*{argmin}_i\ \ d^2(\mathbf{P}^T\mathbf{x},\mathbf{P}^T\mathbf{m}_i) \tag{8}$$

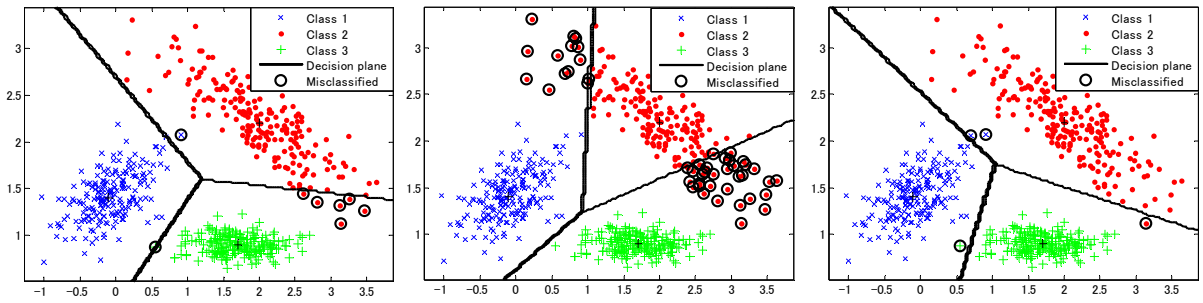
Where $d^2(\mathbf{u},\mathbf{v})$ is the squared Euclidean distance as in:

$$d^2(\mathbf{u},\mathbf{v})=(\mathbf{u}-\mathbf{v})^T(\mathbf{u}-\mathbf{v}) \tag{9}$$

From above equations (5), (6), (7), and (8), we can observe that Fisher optimization criterion is invariant to any scale variation of the transformation vectors; if $\mathbf{p}_i$ is the eigenvector of equation (7), $\alpha_i\mathbf{p}_i$ is also the eigenvector (here we normalize $\|\mathbf{p}_i\|=1$ and only need to tune α_i). All of them are optimal with regard to the Fisher optimization criterion. However, they are not all optimal in the sense of misclassification using the decision function (8). In figure 1 we use 2D toy data to demonstrate this.

In figure1, the 2D toy data consist of 3 classes generated by 3 Gaussian distribution with different scatter matrices and different means. We normalize the length of its LDA vectors $\mathbf{p}_1$ and $\mathbf{p}_2$ to 1.0 first, then by modifying the α_1 and α_2, we can achieve different performances as shown in figure 1(a), and 1(b). It is necessary to learn the optimum norm from the training data. Obviously, Fisher optimization criterion itself can not fulfill this target.

Besides the above problem, the Fisher criteria of LDA mainly rely on class global information such as mean and class scatter instead of local information to find the discriminant directions. Some local-based learning methods such as SVM are local in the sense that solution is exclusively determined by support vectors whereas all other data points are irrelevant to the decision hyperplane



(a) (b) (c)

Figure 1: (a) LDA+NMC, $\alpha_1 = 1.0, \alpha_2 = 1.0$, error rate =1.33%; (b) LDA+NMC, $\alpha_1 = 0.2, \alpha_2 = 1.0$; error rate =7.83%; (c) LDA+RW-LDA+NMC, $\alpha_1 = 3.43, \alpha_2 = 1.81$, error rate =0.67%.

parameters. We hope to learn the optimum norm of LDA vectors by utilizing the local information more effectively. So that the new learned transformation matrix will not only keep the main framework of LDA untouched, but also can achieve better performance using some local information.

3. Re-weighting LDA under Ranking Loss

From above introduction, we knew that the optimum norm of LDA projection can not be determined by Fisher criterion. In this section, we will propose a novel method to decide the optimal norm of LDA vectors after LDA decomposition. We call it reweighting LDA (RW-LDA for short).

In RW-LDA framework, learning the optimum norms of LDA projection vectors is equivalent to finding the transformation $\mathbf{G} = [\alpha_1 \mathbf{p}_1, \cdots, \alpha_h \mathbf{p}_h] \in \mathbb{R}^{H \times h}$ that minimizes loss function under given constraints. Here $\alpha_l \in \mathbb{R}$ and $\mathbf{p}_l$ is the normalized LDA vectors: $\|\mathbf{p}_l\| = 1$. Only α_l need to be estimated.

The squared Euclidean distance of two sample vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{\mathrm{H}}$ in the transformed space is:

$$D(\mathbf{G}^T\mathbf{u}, \mathbf{G}^T\mathbf{v}) = (\mathbf{u}-\mathbf{v})^T \mathbf{G}\mathbf{G}^T (\mathbf{u}-\mathbf{v})$$
$$= \sum_{l=1}^{h} \alpha_l^2 [(\mathbf{u}-\mathbf{v})^T \mathbf{p}_l]^2 \qquad (10)$$

For the multi-class classification problem, in order to minimize the training error (i.e, the number of samples violating decision criterion), we need to supplement the constraints according to the NMC criterion explicitly. In the other word, we try to minimize the training error caused by wrong orders between sample-class pairs. For the sample $\mathbf{x}_{ij}$ of the *i*-th class, the correct distance order between the sample-class pairs should be:

$$\forall k \neq i \qquad D(\mathbf{G}^T\mathbf{x}_{ij}, \mathbf{G}^T\mathbf{m}_k) > D(\mathbf{G}^T\mathbf{x}_{ij}, \mathbf{G}^T\mathbf{m}_i) \qquad (11)$$

In case the classes are not linearly separable, we must allow for misclassifications. To this end, we introduce a non-negative slack variable ξ_{ijk} for each constraint.

$$\forall k \neq i$$
$$D(\mathbf{G}^T\mathbf{x}_{ij}, \mathbf{G}^T\mathbf{m}_k) - D(\mathbf{G}^T\mathbf{x}_{ij}, \mathbf{G}^T\mathbf{m}_i) \geq 1 - \xi_{ijk} \qquad (12)$$
$$\xi_{ijk} \geq 0 \qquad (13)$$

The above constraint imposes a margin between the classes similarly to the margin defined for the SVMs (also it can be considered as the hinge loss). Just as the hinge loss in SVMs is only triggered by examples near the decision boundary, the hinge loss in equation (12) is only triggered by differently labeled examples that invade other class's neighborhood.

Note for sample $\mathbf{x}_{ij}$, take

$$\gamma_{ij} = \max_k \xi_{ijk} . \qquad (14)$$

Then

If $1 \leq \gamma_{ij}$, $\mathbf{x}_{ij}$ is misclassified;

If $0 < \gamma_{ij} < 1$, $\mathbf{x}_{ij}$ is correctly classified, however it violates the margin of the decision function;

If $\gamma_{ij} = 0$, $\mathbf{x}_{ij}$ is correctly classified;

So $\sum \gamma_{ij}$ is the upper bound of the number of misclassified samples on the training set. To minimize this upper bound, we can effectively control the misclassification over training set.

On the other hand, if all constraints are feasible, the solution is typically not unique. We aim to select the solution such that utilizes as few features as possible. Following [14, 19], we minimize the sum of squared norm of α_l. By substituting $w_l = \alpha_l^2$, the **RW-LDA** framework leads to the following optimization problem:

$$\underset{w_l, \gamma_{ij}}{\operatorname{argmin}} \sum_{l=1}^{h} w_l + \mu \sum_{i,j} \gamma_{ij} \qquad (15)$$

subject to: $\forall k \neq i$

$$D(\mathbf{G}^T\mathbf{x}_{ij}, \mathbf{G}^T\mathbf{m}_k) - D(\mathbf{G}^T\mathbf{x}_{ij}, \mathbf{G}^T\mathbf{m}_i) \geq 1 - \xi_{ijk}, \quad (16)$$

$$\xi_{ijk} \geq 0, \quad (17)$$

$$\gamma_{ij} = \max_k \xi_{ijk}, \quad (18)$$

$$w_l \geq 0, \quad l = 1, \cdots, h \quad (19)$$

In above objective function, as introduced, the first term ($\sum_l w_l$) penalizes the norm of the parameter to use as few features as possible, while the other ($\sum_{ij} \gamma_{ij}$) incurs the hinge loss for examples that violate the condition of unit margin to minimize the number of misclassifications. And μ is a constant to tradeoff the two terms. The cost function thereby favors distance metrics in which differently labeled samples maintain a large margin of distance and do not threaten to "invade" other class's neighborhoods.

The optimization of RW-LDA can be considered as a special case of ranking-based learning methods. In general ranking-based learning methods, such as RankBoost [24], Ranking SVM [25], they try to minimize the number of wrong orders in instance sets (such as the movie-ranking applications) or instance pairs (such as the webpage retrievals). While in our RW-LDA, we try to minimize the wrong orders of sample-class pairs; in other words, we try to keep the distance of the samples to its correct class centers not to exceed the distances to any other class centers. In section 4, we will point out the similarities and differences between them in detail.

So using the above reweighting LDA framework we can not only keep the original Fisher optimum criterion untouched, but also achieve the better performance using local information to tune the optimum norm. In figure 1(c), we can see that after RW-LDA, we get $\alpha_1 = 3.43$, and $\alpha_2 = 1.81$; and the error rate can be reduced from original 1.33% to 0.67%.

3.1. Optimization of RW-LDA Algorithm

As equations (15)-(19) show, the original optimization problem can be reformulated as an instance of quasi-convex optimization problems [28]. The objective function and all constraints are linear except constraint (18) which is nonlinear and quasi-convex. Besides this, constraint (19) requires all unknown variables of norms to be non-negative. And that is why we can not directly utilize some proposed efficient optimization methods [30] for multi-class SVM by decomposing the large scale convex and quadratic optimization problem into small problems.

Although this quasi-convex optimization can be solved by standard online packages, general-purpose solvers tend to scale poorly in the number of constraints. Thus, for our work, we implemented our own special-purpose solver,

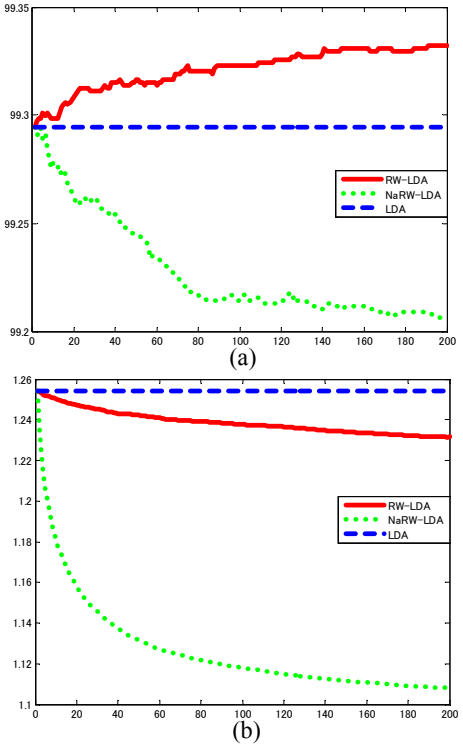


Figure 2: The curve of (a) Rank-1 performance (%); (b) average rank performance with the optimization iterations (horizontal axle) for RW-LDA, NaRW-LDA and LDA

exploiting the fact that most of the slack variables ξ_{ijk} and γ_{ij} never attain positive values. The slack variables are sparse because most sample-class pairs are well separated; thus, their resulting pairwise distances do not incur the hinge loss, and we obtain very few active constraints.

Our solver was based on a combination of gradient descent and alternating projection methods [26]. We projected updates in w_l back onto the positive semi-definite cone after each step to confirm $w_l \geq 0$ hold. Our implementation worked much faster than generic solvers, and could deal with more than several hundreds of thousands of training samples with several thousands of classes. For example, in the experiment of section 3.2 consisting of 77000 face samples from 2500 persons, the training time using our optimization method for 200 iterations to attain stable result is about 2600 seconds on PIV 3.4GHz PC.

3.2. Discussion

Inspired by previous work on LESS (Lowest Error in a Sparse Subspace) models [14, 15], we also can optimize the norm of LDA vectors using a variant of objective function

(15), which leads to the following algorithm (we call it Naïve RW-LDA, NaRW-LDA for short):

$$\underset{w_l,\xi_{ijk}}{\operatorname{argmin}} \sum_{l=1}^{h} w_l + \mu \sum_{i,j,k} \xi_{ijk} \tag{20}$$

subject to: $\forall i \neq k$

$$D(\mathbf{G}^T\mathbf{x}_{ij}, \mathbf{G}^T\mathbf{m}_k) - D(\mathbf{G}^T\mathbf{x}_{ij}, \mathbf{G}^T\mathbf{m}_l) \geq 1 - \xi_{ijk} \tag{21}$$

$$\xi_{ijk} \geq 0 \tag{22}$$

$$w_l \geq 0\ ,\ \ l = 1, \cdots, h \tag{23}$$

The difference between RW-LDA and NaRW-LDA is obvious. In RW-LDA, every sample only brings one slack at most to the objective function using $\gamma_{ij} = \max_k \xi_{ijk}$, while in NaRW-LDA, every sample-class pair constraint may bring one slack to the objective function. As the result, NaRW-LDA pays more attention to minimize the total number of violated sample-class pair constraints instead of the total number of mis-classified samples. In fact, the total number of mis-classified samples is related to the Rank-1 performance, and the number of violated sample-class pair constraints is related to the average rank performance (because for one sample $\mathbf{x}_{ij}$, the number of violated sample-class pair distance constraints is equal to the order of $\mathbf{m}_i$ in candidate list minus 1). So for the multi-class classification problems, the optimization criterion of RW-LDA is more favorable than that of NaRW-LDA.

We compared the above two methods in face recognition. The face data set consists of total 77000 samples from 2500 persons. Original feature dimension is 3200; PCA is first utilized to reduce dimension to 1000 to cover 95% of variance, and LDA is utilized next to reduce feature dimension to 200 further; based on this, we utilize RW-LDA and NaRW-LDA separately. Figure 2 gives the curve of Rank-1 performance and average rank of RW-LDA, NaRW-LDA and LDA with the number of iterations of optimization. We can see although the NaRW-LDA can effectively improve the average rank performance, it can not reduce Rank-1 errors more effectively than RW-LDA.

4. Related Works

In this section we briefly review some recent methods in the field of LDA-related variants, metric learning under relative distance constraints and ranking-based learning, pointing out similarities and differences with our work.

As we know classical LDA transformation is non-orthogonal. Ye *et al*. [6, 23] proposed an orthogonal LDA (OLDA) algorithm to find orthogonal basis using QR decomposition under Fisher criterion. Our method is similar to OLDA in that we also seek the optimum LDA transformation. However, OLDA seeks for the optimum orthogonal transformation instead of the optimum norm of the LDA vectors, and their optimum criteria is still the Fisher criterion and different from ours. Sparse LDA [21] was another recently proposed method related to LDA. Sparse LDA keeps the original LDA framework untouched with one additional constraint to minimize the number of the non-zero elements of LDA vectors instead of the optimum norm of the LDA vectors.

From the viewpoint of the metric learning, the optimization of the norm of LDA vectors can be considered as learning a diagonal metric in the LDA transformed space. Veenman *et al*. [14, 15] proposed LESS (Lowest Error in a Sparse Subspace) algorithm to learn the optimum diagonal metric using nearest mean classifier. The differences between them are that our diagonal metric is learned in the LDA-transformed space, and their learned optimum diagonal metric has no relation with LDA at all; LESS only deals with binary classification problems. The natural extension of LESS to multi-class classification problems leads to NaRW-LDA. Finally, the optimization of LESS is different from ours, in which we use gradient based method instead of their linear programming method, and we can deal with about several hundreds of thousands of training samples with several thousands of classes, and their method only can deal with several thousands of samples with hundreds of classes.

Schultz *et al.* [12] proposed an online learning algorithm for learning a Mahalanobis distance metric. The metric is trained with the goal that all similarly labeled inputs have small pairwise distances (bounded from above), while all differently labeled inputs have large pairwise distances (bounded from below). A margin is defined by the difference of these thresholds and induced by a hinge loss function. Similar with this, Weinberger *et al*. [13] proposed a distance metric learning methods to maximize the maximum margin between nearest neighbor (LMNN). Our work has a similar basis with above works in its appeal to margins and hinge loss functions, but differs in the way to deal with multi-class classification. In particular, we do not seek to minimize the distance between all similarly labeled neighbored samples, and do not minimize the slack variables incurred by every violated sample-class or sample-sample pair constraint. And we focus on learning the metric in the LDA-transformed space.

Ranking SVMs proposed by Joachims *et al*. [25] is constructed on a quadratic risk minimization framework with the goal to minimize the number of mis-orderings between the predicted ranks and target ranks, similar to the common classification SVMs. Qin *et al*.[16] proposed a multiple hype-plane based on Ranking SVM. RankBoost [24] utilizes the gradient-based optimization method to minimize the exponential risk of wrong orders between sample-sample pairs. In essence, aforementioned rank learning algorithms transform ranks into a set of pairwise

relationships between samples and thus cast it into a classification problem. Thus, our method differs in the forms of the optimization problems and also the optimization method. We do not require all relevant distances should be smaller than all irrelevant distance like RankSVM does. And our optimization risk upper bound is also different from that of RankBoost.

5. Experiments

To evaluate the performance of proposed RW-LDA algorithm, some classification experiments are performed on the following types face images.

The methods used in the following comparisons include:

Ed-NMC: nearest mean classifier based Euclidean distance;

PCA: Principal component analysis;

LDA: Linear discriminant analysis (the norms of all *LDA* vectors are normalized to 1.0, so is the following *NDA*);

NDA: Nonparametric discriminant analysis [10];

OLDA: Orthogonal linear discriminant analysis [6, 23]:

RW-LDA: Reweighting LDA;

NaRW-LDA: Naive Reweighting LDA;

RankBoost: RankBoost uses *LUT*-based (Look Up Table) classifier [31] as the weak ranker, *LUT* weak ranker is constructed based on the absolute feature difference between sample-sample pairs, and the number of bins for each LUT weak ranker is 20 according to experience.

The results of experiments are reported separately.

5.1. Experiment on ORL Face DB

The ORL face recognition data set contains 400 grayscale images of 40 individuals in 10 different poses. We randomly selected 3 images of each person for training and 7 images for testing. For every facial image, 3200-dimension Gabor feature was extracted to code its appearance. To overcome small-sample-size problem, all the algorithms are preceded by *PCA* and then performed in the transformed 80-dimensional *PCA* subspace which accounts for 95% of total variance. The experiments are repeated 100 times. All contrastive experiments are based on the identical partition of the training/test set for each dataset. The performances given in table 1 are the average value on 100 experiments.

From the table 1, we can confirm that *RW-LDA* can generally improve the performance by finding the optimum norms of vectors no matter *LDA*, *OLDA* or *NDA*. For example, by utilizing *RW-LDA*, the error rate of framework *PCA+LDA+Ed-NMC* can be reduced by about 1/4, from 3.91% to 2.90%. And the comparison between *PCA+LDA+NaRW-LDA+Ed-NMC* and *PCA+LDA+ RW-LDA+Ed-NMC* also proves that *RW-LDA* algorithm is more effective to improve the Rank-1 performance compared with *NaRW-LDA*. For the experiment of RankBoost, the number of weak ranker is decided to be 10 according to experience. Although on training set, the TOP1 training error of RankBoost is 0.0. Its generalization ability on test set is not very good.

On ORL data set, because the number of training samples is few (120 samples) and feature dimension after *LDA* is not high (39 dimension), the training time for *RW-LDA* is about 4s using Matlab on PIV 3.4G CPU.

Table 1 Face recognition error rates on ORL databases (%).

Method	Error rate on test set (%)	
	mean	std
PCA+Ed-NMC	11.99	2.18
PCA+LDA+Ed-NMC	3.91	1.80
PCA+LDA+NaRW-LDA +Ed-NMC	4.33	2.49
PCA+LDA+RW-LDA +Ed-NMC	**2.90**	**1.60**
PCA+OLDA+Ed-NMC	4.03	1.73
PCA+OLDA+RW-LDA+ Ed-NMC	3.71	1.82
PCA+NDA+Ed-NMC	6.58	1.98
PCA+NDA+RW-LDA+ Ed-NMC	5. 98	1.85
RankBoost	6.41	1.95

5.2. Experiment on FERET Face DB

The FERET evaluation protocol was designed to measure performance on different galleries and probe sets for identification and verification tasks [29]. In our experiments we use the FERET training set to learn discriminant functions for the different algorithms, which are then evaluated with the FERET probe sets and gallery set. The FERET training set consists of 1002 images from 429 persons, while the gallery consists of 1196 distinct individuals with one image per individual. Probe sets are divided into four categories: the Fa/Fb set consists of 1195 frontal images; the Fa/Fc set consists of 194 images taken with a different camera under different lighting on the same day as the corresponding gallery image; Duplicate 1 contains 722 images that were taken on different days within one year from the acquisition of the probe image and corresponding gallery image; Duplicate 2 contains 234 images that were taken on different days at least one year apart. The face recognition experiment on this database is different from above because it is an open-set verification problem.

Similar to the experiment on ORL, for every facial image, 3200-dimension Gabor feature was extracted to code its appearance. *PCA* was utilized to reduce the feature dimension to 500 which accounts for 95% of total variance. Then in the in the transformed *PCA* subspace and *LDA*,

OLDA or *NDA* were utilized to reduce the dimension to 200.

Table 2 Face recognition error rates on FERET databases (%).

Method	Fa/fb	Fa/fc	Dup1	Dup2
PCA+Ed-NMC	10.88	71.13	50.97	78.21
PCA+LDA+Ed-NMC	1.59	39.69	39.20	64.53
PCA+LDA+RW-LDA +Ed-NMC	**0.92**	**31.93**	38.09	64.96
PCA+OLDA+Ed-NMC	1.42	37.11	38.70	59.50
PCA+OLDA+RW-LDA +Ed-NMC	1.22	35.56	**34.07**	**54.70**
PCA+NDA+Ed-NMC	1.34	41.24	38.23	61.97
PCA+NDA+RW-LDA +Ed-NMC	1.30	42.27	38.78	61.11
RankBoost	2.85	63.20	43.32	72.64

From table 2, we can observe the similar trend with the result on ORL. However, compared with *LDA* and *OLDA*, the improvement on *NDA* using *RW-LDA* is not so obvious. Sometimes, the performance of *PCA+NDA+RW-LDA +Ed-NMC* is even worse than *PCA+NDA+ Ed-NMC*. That might be due to that *NDA* also utilizes samples from nearest neighbors (local information) to construct the between-class and with-in class scatter matrix, which is similar to our *RW-LDA* to some degree. For the experiment of *RankBoost*, it still does not exhibit comparable performance. It might be due to that the simple absolute feature difference can not reflect the similarity of sample-sample pair effectively, and there are more extra parameters to be tuned.

Fig. 3 illustrates the improvements more graphically by showing how the Rank-1 matches change as a result of *RW-LDA* to learn the optimum norms of *LDA* vectors on test set, from a mismatch using *PCA+LDA* to a match using *PCA+LDA+RW-LDA*. (Though the algorithm operated on Gabor wavelet feature, for clarity the figure shows the original clipped images).

5.3. Comparison with Normalization Methods

In some way, adjusting the norm of the *LDA* vectors is equivalent to normalizing the data. *RW-LDA* can be considered as a supervised normalization method. In practice, normalizing the data appropriately could have an important impact on the overall performance. The popular normalization methods include the normalization of mean and variance (*MV*-norm), and the normalization of vector length (*VL*-norm) etc. Generally these normalization methods do not utilize the class information (unsupervised). This is the main difference between normalization methods and *RW-LDA*. Besides this, normalization methods

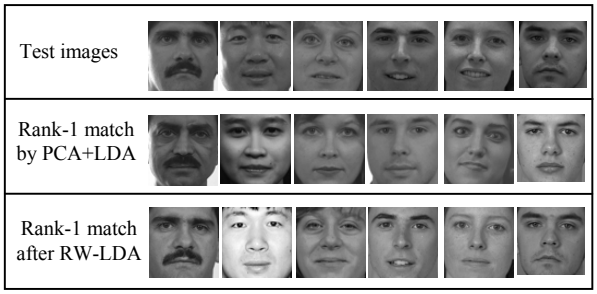


Figure 3: Images from the FERET face recognition database. *Top row*: some test images correctly recognized by PCA+ LDA +RW-LDA, but not PCA+LDA. *Middle row*: the corresponding incorrect rank-1 match using PCA+LDA. *Bottom row*: the correct rank-1 match after using RW-LDA.

themselves can not solve the problem of *LDA* completely when the classes do not share the same distribution.

We compare the *RW-LDA* with these normalization methods using the same toy data in figure 1. It is can be seen obviously that even applying normalization beforehand, there is still space to improve the performance of *LDA* by utilizing *RW-LDA*. For example, if we first normalize the vector lengths of data, then utilize *LDA*, and the error rate is 1.0%; if we apply *RW-LDA* after *LDA*, the error rate can be reduced to 0.33% further.

Table 3 Error rates using different normalization methods on toy data.

Method	Error rate (%)
LDA+Ed-NMC	1.33
LDA+RW-LDA+Ed-NMC	0.67
MV-Norm+LDA+Ed-NMC	2.0
MV-Norm+LDA+RW-LDA+Ed-NMC	0.67
VL-Norm+LDA+Ed-NMC	1.0
VL-Norm+LDA+RW-LDA+Ed-NMC	**0.33**

6. Conclusions

In this paper, we have shown how to learn the optimum norm of LDA vectors under the ranking loss for nearest mean classifier by Quasi-convex programming. Our framework makes no assumptions about the structure or distribution of the data and scales naturally to large number of classes. Experimental results demonstrate the effectiveness of the proposed RW-LDA algorithm not only for LDA, but also for NDA and other LDA variations. Ongoing work is focused in several directions. First, we are working to learn general Mahalanobis metric instead of only diagonal metric in LDA-transfomed space. Second,

we are investigating the kernel trick to perform RW-LDA classification in nonlinear feature spaces combing with Kernel LDA [27]. Finally, we are extending our framework to learn the optimum norm of LDA vectors utilizing nearest neighbor class under normalized correlation metric instead of nearest mean classifier under Euclidean distance. Such framework should lead to even more flexible and powerful classifiers.

Acknowledgments

This paper was partially supported by Ministry of Economy, Trade, and Industry of Japan under the project - Development of Intelligence Technology for the Next Generation Robots.

References

[1] R. Duda, P. Hart, & D. Stork. Pattern classification. Wiley, 2000.

[2] K. Fukunaga. Introduction to statistical pattern classification. San Diego, California, USA: Academic Press. 1990.

[3] T. Hastie, R. Tibshirani, and J. Friedman. The elements of statistical learning: data mining, inference, and prediction. Springer. 2001.

[4] P. Belhumeour, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Trans Pattern Analysis and Machine Intelligence, 19: 711–720, 1997.

[5] S. Dudoit, J. Fridlyand,, and T. P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. Journal of the American Statistical Association, 97, 77–87. 2002.

[6] J. Ye. Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems. Journal of Machine Learning Research, 6: 483–502. 2005.

[7] M. Loog, R. P. W. Duin. Linear dimensionality reduction via a heteroscedastic extension of LDA: the Chernoff criterion. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 26 (6): 732-739, 2004.

[8] X.Qiu, and L.D.Wu. Face recognition by stepwise nonparametric margin maximum criterion. IEEE International Conference on Computer Vision, Vol.2: 1567-1572, 2005.

[9] J. Friedman. Regularized discriminant analysis. Journal of the American Statistical Association, 84: 165–175, 1989.

[10] Z. Li, W. Liu, D. Lin, and X. Tang. Nonparametric subspace analysis for face recognition. IEEE Conf. on Computer Vision and Pattern Recognition Vol.2, 961-966, 2005.

[11] L. Chen, H. Liao, M. Ko, J. Lin, and G. Yu. A new LDA-based face recognition system which can solve the small sample size problem. Pattern Recognition, 33, 1713–1726. 2000

[12] M. Schultz, and T. Joachims. Learning a distance metric from relative comparisons. Advances in Neural Information Processing Systems, 2004

[13] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. Advances in Neural Information Processing Systems 18. MIT Press: Cambridge, MA 2006.

[14] C. J. Veenman, and D.M.J. Tax. A weighted nearest mean classifier for sparse subspaces. IEEE Conference on Computer Vision and Pattern Recognition, 2005.

[15] C. J. Veenman, and D.M.J. Tax. LESS: A model-based classifier for sparse subspaces. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(9), 2005.

[16] T. Qin, T.Y. Liu, W. Lai, X. D. Zhang, and H. Li. Ranking with multiple hyper-planes. Proceeding of the 30th International ACM SIGIR Conference, 2007.

[17] J. Gao, H. Qi, X. Xia, and J.Y. Nie, Linear discriminant model for information retrieval, Proceedings of the 28th international ACM SIGIR conference, 2005, pp. 290–297.

[18] H. F. Li, T. Jiang, and K.S. Zhang. Efficient and robust feature extraction by maximum margin criterion. In Proc. of Neural Information Processing Systems, 2003.

[19] I. W. Tsang and J.T. Kwok. Distance metric learning with kernels. Proceedings of the International Conference on Artificial Neural Networks, 2003.

[20] R. Yan and A. G. Hauptmann. Efficient margin-based rank learning algorithms for information retrieval. International Conference on Image and Video Retrieval, July 13-15, 2006

[21] B. Moghaddam, Y. Weiss, and S. Avidan. Generalized spectral bounds for sparse LDA. Proceedings of the 23rd international conference on Machine learning, vol.148: 641–648, 2006.

[22] T. Hastie, A.Buja, and R. Tibshirani. Penalized discriminant analysis. Annals of Statistics, 23: 73–102, 1995.

[23] J. P. Ye, and T. Xiong. Null space versus orthogonal linear discriminant analysis. Proceedings of the 23rd international conference on Machine learning, vol.148: 1073-1080, 2006.

[24] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. The Journal of Machine Learning Research archive, 4: 933-969, 2003.

[25] T. Joachims, Optimizing search engines using clickthrough data, Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), ACM, 2002.

[26] L. Vandenberghe and S. P. Boyd. Semidefinite programming. SIAM Review, 38(1):49–95, March 1996.

[27] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.R. Muller, Fisher discriminant analysis with kernels, Neural Networks for Signal Processing IX, pp. 41-48, 1999.

[28] R. Hartley, F. Kahl. Tutorial on continuous optimization, European Conference on Computer Vision, 2006.

[29] P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rizvi, and Patrick J. Rauss. The FERET evaluation methodology for face-recognition algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence Vol.12, No.10, (2000) 671-678

[30] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. Journal of Machine Learning Research, 2:265--292, 2001.

[31] C. Huang, H.Z. Ai, B. Wu, and S.H. Lao Boosting nested cascade detector for multi-view face detection. Proceedings of the 17th International Conference on Pattern Recognition, pp. 415-418 Vol.2. 2004.