Oriented Discriminant Analysis (ODA)

Fernando De la Torre **Takeo Kanade**
ftorre@cs.cmu.edu tk@cs.cmu.edu

Robotics Institute, Carnegie Mellon University,
Pittsburgh, Pennsylvania 15213.

Abstract

Linear discriminant analysis (LDA) has been an active topic of research during the last century. However, the existing algorithms have several limitations when applied to visual data. LDA is only optimal for gaussian distributed classes with equal covariance matrices and just classes-1 features can be extracted. On the other hand, LDA does not scale well to high dimensional data (over-fitting) and it does not necessarily minimize the classification error. In this paper, we introduce Oriented Discriminant Analysis (ODA), a LDA extension which can overcome these drawbacks. Three main novelties are proposed:

- *An optimal dimensionality reduction which maximizes the Kullback-Liebler divergence between classes is proposed. This allows us to model class covariances and to extract more than classes-1 features.*
- *Several covariance approximations are introduced to improve classification in the small sample case.*
- *A linear time iterative majorization method is introduced in order to find a local optimal solution.*

Several synthetic and real experiments on face recognition are reported [1].

1 Introduction

Canonical Correlation Analysis (CCA) , Independent Component Analysis (ICA), Linear Discriminant Analysis (LDA), Principal Component Analysis (PCA), ... are some examples of subspace methods (SM) useful for classification, dimensionality reduction and data modeling. These methods have been actively researched by the statistics, neural networks, machine learning and vision communities during the last century. In particular, SM have been very successful in computer vision to solve problems like structure from motion [16] or detection/recognition [17]. SM can be especially useful when available data increases in features/samples, since there is a need for dimensionality reduction while preserving relevant attributes of the data[2]. Another benefit of many subspace methods is that they can be computed as an eigenvalue problem, for which there are efficient numerical packages. A drawback of SM is its linear assumption, however kernel methods and latent variable models have made recover the interest.

[1] This work has been partially supported by National Business Center of the Department of the Interior under a subcontract from SRI International, U.S. Department of Defense contract N41756-03-C4024, NIMH Grant R01 MH51435 and DARPA HumanID program under ONR contract N00014-00-1-0915.

[2] Also many times it is helpful to find a new coordinate system (e.g. Fourier transform).

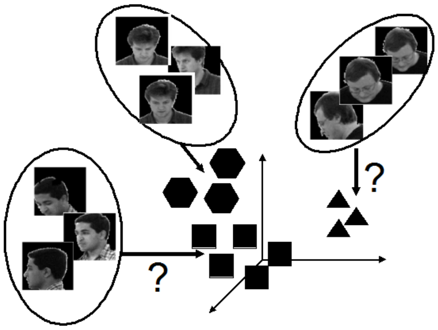


Figure 1: Face classification by projecting onto a low dimensional space.

In particular, LDA has been extensively used for classification problems such as speech recognition, face recognition/tracking [19] or multimedia information retrieval [3, 2, 5, 7, 20, 21, 13]. Among several classification methods (e.g. Support Vector Machines, decision trees, ...), LDA remains a powerful preliminary tool for dimensionality reduction to preserve discriminative features, avoid the ”curse of dimensionality” and understanding better the data (e.g visualization). This is especially important in the context of computer vision, where usually high dimensional data is present and a preliminary dimensionality reduction is often necessary.

However, there are several issues which remain unsolved when applying LDA to high dimensional data (e.g. images). LDA is only optimal in the case that all the classes have a gaussian distribution with equal covariance matrices, due to this restriction, the maximum number of features that can be extracted is just the number of classes -1. Another problem of LDA is the small size problem [20, 21]. In the case that we have more ”dimensions”[3] than data samples, LDA overfits the data due to bad estimates of the covariance matrices and PCA techniques usually outperform LDA [13]. Also, in this case the computational and storage requirements of traditional algorithms do not scale well. In this paper we introduce Oriented Discriminant Analysis (ODA), a new low dimensional discriminatory technique which is able to solve previous LDA problems. Figure 1 illustrates the main purpose of this paper[4].

2 Linear Discriminant Analysis

The aim of discriminant analysis methods is to project the data from several classes into a subspace of lower dimension, so that the classes are as compact and they are as far as possible from each other. In particular, LDA remains a powerful tool for dimensionality while extracting features which preserve class separability.

Several optimization criteria for LDA are possible and most of them are based on

[3] In this case the true dimensionality of the data is the number of samples.

[4] Bold capital letters denote a matrix $\mathbf{D}$, bold lower-case letters a column vector $\mathbf{d}$. $\mathbf{d}_j$ represents the j column of the matrix $\mathbf{D}$. All non-bold letters will represent variables of scalar nature. *diag* is an operator which transforms a vector to a diagonal matrix. $\mathbf{1}_k \in \Re^{k \times 1}$ is a vector of ones. $\mathbf{I}_k \in \Re^{k \times k}$ is the identity matrix and $\mathbf{e}_i$ is the i columun. $tr(\mathbf{A}) = \sum_i a_{ii}$ is the trace of the matrix $\mathbf{A}$. $||\mathbf{A}||_F = tr(\mathbf{A}^T\mathbf{A}) = tr(\mathbf{A}\mathbf{A}^T)$ designates the Frobenious norm of a matrix. $N_d(\mathbf{x}; \boldsymbol{\mu}, \mathbf{\Sigma})$ indicates a d-dimensional gaussian on the variable $\mathbf{x}$ with mean $\boldsymbol{\mu}$ and covariance $\mathbf{\Sigma}$.

relations between the following covariance matrices, which can be conveniently expressed in matrix form as:

$$\mathbf{S}_t = \frac{1}{n-1}\mathbf{D}\mathbf{P}_1\mathbf{D}^T \quad \mathbf{S}_w = \frac{1}{n-1}\mathbf{D}\mathbf{P}_2\mathbf{D}^T \quad \mathbf{S}_b = \frac{1}{n-1}\mathbf{D}\mathbf{P}_3\mathbf{D}^T \tag{1}$$

where $\mathbf{D} \in \Re^{d\times n}$ is the data matrix. $\mathbf{P}_i$ are projection matrices (i.e $\mathbf{P}_i^T = \mathbf{P}_i$ and $\mathbf{P}_i^2 = \mathbf{P}_i$) with the following expressions:

$$\mathbf{P}_1 = \mathbf{I} - \tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^T \quad \mathbf{P}_2 = \mathbf{I} - \mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T \qquad \mathbf{P}_3 = \mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T - \frac{1}{n}\mathbf{1}_n\mathbf{1}_c\mathbf{G}^T \tag{2}$$

where $\mathbf{G} \in \Re^{n\times c}$ is an dummy indicator matrix such that $\sum_j g_{ij} = 1$, $g_{ij} \in \{0,1\}$ and g_{ij} is 1 if $\mathbf{d}_i$ belongs to class j. c denotes the number of classes and n the number of samples. $\mathbf{S}_b$ is the between covariance matrix and represents the average of the distances between the mean of the classes. $\mathbf{S}_w$ represents the within covariance matrix and it is a measure of the average compactness of each class. Finally $\mathbf{S}_t$ is the total covariance matrix. With the matrix expressions it is straightforward to show that $\mathbf{S}_t = \mathbf{S}_w + \mathbf{S}_b$. The upper bounds on the ranks of the matrices are $c-1$, $n-c$, $n-1$ for $\mathbf{S}_b, \mathbf{S}_w, \mathbf{S}_t$ respectively.

A Rayleigh quotient like are among the most popular LDA optimization criteria [7], some are: $J_1(\mathbf{B}) = \frac{|\mathbf{B}^T\mathbf{S}_1\mathbf{B}|}{|\mathbf{B}^T\mathbf{S}_2\mathbf{B}|}$ $J_2(\mathbf{B}) = tr((\mathbf{B}^T\mathbf{S}_1\mathbf{B})^{-1}\mathbf{B}^T\mathbf{S}_2\mathbf{B})$ $J_3(\mathbf{B}) = \frac{tr(\mathbf{B}^T\mathbf{S}_1\mathbf{B})}{tr(\mathbf{B}^T\mathbf{S}_2\mathbf{B})}$, where $\mathbf{S}_1 = \{\mathbf{S}_b, \mathbf{S}_b, \mathbf{S}_t\}$ and $\mathbf{S}_2 = \{\mathbf{S}_w, \mathbf{S}_t, \mathbf{S}_w\}$. Although other constrained optimization formulations are possible [4, 7]. A closed form solution to previous minimization problems is given by a generalized eigenvalue problem $\mathbf{S}_1\mathbf{B} = \mathbf{S}_2\mathbf{B}\Lambda$. The generalized eigenvalue problem can be solved as a joint diagonalization, that is, finding a common basis $\mathbf{B}$ which diagonalizes simultaneous both matrices $\mathbf{S}_1$ and $\mathbf{S}_2$ (i.e. $\mathbf{B}^T\mathbf{S}_2\mathbf{B} = \mathbf{I}$ and $\mathbf{B}^T\mathbf{S}_1\mathbf{B} = \Lambda$).

3 Oriented Discriminant Analysis

LDA is the optimal discriminative projection only in the case of having gaussian classes with equal covariance matrix [1, 5] (assuming enough training data). LDA will not be optimal if the classes have different covariances. Fig. 2 shows one situation where two classes have almost orthogonal principal directions of the covariances and close means. In this pathological case LDA chooses the worse possible discriminative direction where the classes are overlapped (it is also very numerically unstable), whereas ODA finds a better projection. In general, this situation becomes increasingly dangerous when we increment the number of classes and the classes are closer (in terms of means and covariances).

In order to relax this problem, several authors have proposed extensions and new views of LDA. Campbell [1] derives a maximum likelihood approach to discriminant analysis by assuming all the classes have equal covariance matrix, he shows that LDA is equivalent to impose that the class means lie in a l-dimensional subspace. Following this approach, Kumar and Andreou [11] proposed heteroscedastic discriminant analysis where they incorporate the estimation of the means and covariances in the low dimensional space. On the other hand, Saon et *al.* [15] define a new energy function to model the directionality of the data, $J(\mathbf{B}) = \prod_{i=1}^{c}(\frac{|\mathbf{B}^T\mathbf{S}_b\mathbf{B}|}{|\mathbf{B}^T\boldsymbol{\Sigma}_i\mathbf{B}|})^{n_i}$, where $\boldsymbol{\Sigma}_i$ is the class covariance matrix and $\mathbf{S}_b$ the between-class scatter covariance matrix. In this paper we extend previous approaches by deriving a probabilistic interpretation of the optimal discriminant analysis in the case of having classes with different covariances. Also, our method scales well to high dimensional data and efficient algorithms are developed.

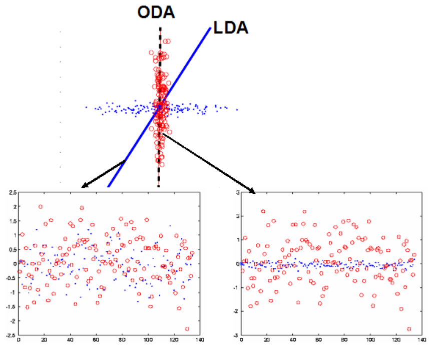


Figure 2: Projection onto LDA direction and ODA.

3.1 Maximizing Kullback-Leibler distance.

In order to take into account the class covariance information and assuming that the classes are gaussians, in this section we will derive the optimal linear dimensionality reduction which maximizes a distance between the projected classes.

A simple measure of distance between two gaussian distributions $N(\mathbf{x};\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)$ and $N(\mathbf{x};\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)$ is given by the Kullback-Leibler (KL) divergence [7]:

$$KL_{ij} = \tfrac{1}{2}\int d\mathbf{x}\big(N(\mathbf{x};\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i) - N(\mathbf{x};\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)\big)\, log\tfrac{N(\mathbf{X};\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)}{N(\mathbf{X};\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)}$$
$$= \tfrac{1}{2}tr(\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\Sigma}_j + \boldsymbol{\Sigma}_j^{-1}\boldsymbol{\Sigma}_i - 2\mathbf{I}) + \tfrac{1}{2}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T(\boldsymbol{\Sigma}_j^{-1} + \boldsymbol{\Sigma}_i^{-1})(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) \quad (3)$$

The Kullback-Leibler distance between two gaussian distributions is proportional to the distance between their means weighted by their covariances.

We would like to find a linear transformation $\mathbf{B}$, common to all the classes (i.e. $N(\mathbf{B}^T\boldsymbol{\mu}_i, \mathbf{B}\boldsymbol{\Sigma}_i\mathbf{B}^T)$ such that it maximizes the separability between the classes in the low dimensional space, that is, we want to maximize $E(\mathbf{B}) = \sum_{i=1}^c\sum_{j=1}^c tr\big((\mathbf{B}^T\boldsymbol{\Sigma}_i\mathbf{B})^{-1}(\mathbf{B}^T\boldsymbol{\Sigma}_j\mathbf{B}) + (\mathbf{B}^T\boldsymbol{\Sigma}_j\mathbf{B})^{-1}(\mathbf{B}^T\boldsymbol{\Sigma}_i\mathbf{B})\big) + (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T\mathbf{B}((\mathbf{B}^T\boldsymbol{\Sigma}_j\mathbf{B})^{-1} + (\mathbf{B}^T\boldsymbol{\Sigma}_i\mathbf{B})^{-1})\mathbf{B}^T(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)$. After some simple algebraic arrangements, the previous equation can be expressed in a more compact and enlighting manner:

$$G(\mathbf{B}) = -\sum_{i=1}^{c} tr\big((\mathbf{B}^T\boldsymbol{\Sigma}_i\mathbf{B})^{-1}(\mathbf{B}^T\mathbf{A}_i\mathbf{B})\big) \quad \mathbf{A}_i = \sum_{j\neq i}^{c}\big((\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T + \boldsymbol{\Sigma}_j\big) \quad (4)$$

Observe that we have introduced a negative sign for convenience, rather than searching for a maximum, we will be interested in finding a minimum of $G(\mathbf{B})$. $\mathbf{A}_i = \mathbf{M}\mathbf{P}_i\mathbf{M}^T + \sum_{j\neq i}^c\boldsymbol{\Sigma}_j$, where $\mathbf{M} \in \mathbf{R}^{d\times c}$ is a matrix such that each column is the mean of each class and $\mathbf{P}_i = \mathbf{I}_c + c\mathbf{e}_i\mathbf{e}_i^T - \mathbf{e}_i\mathbf{1}_c^T - \mathbf{1}_c\mathbf{e}_i^T \in \mathbf{R}^{c\times c}$. Several interesting things are worth pointing out from eq. 4. If all covariances are the same (i.e. $\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}\ \forall i$), eq. 4 results in $tr\big((\mathbf{B}^T\boldsymbol{\Sigma}\mathbf{B})^{-1}(\mathbf{B}^T\sum_{i=1}^c\sum_{j\neq i}^c(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T\mathbf{B})\big) + c(c-1)l$, this is exactly what LDA maximizes. ODA takes into account not just the distance between the means but also the orientation and magnitude of the covariance. In the LDA case, the number of extracted features can not exceed the number of classes, because the rank of $\mathbf{S}_b$ is $c-1$, however in ODA we do not have this constraint and more features can be obtained. Unfortunately, due to different normalization factors $(\mathbf{B}^T\boldsymbol{\Sigma}_i\mathbf{B})^{-1}$, eq. 4 does not have a closed form solution in terms of an eigenequation (not an eigenvalue problem).

4 Bound optimization

Eq. 4 is hard to optimize, second order type of gradient methods (e.g. Newton or conjugate gradient) do not scale well with huge matrices (e.g. $\mathbf{B} \in \Re^{d \times l}$). Moreover, in this particular energy function the second derivative is quite complex. In this section we introduce a bound optimization method called iterative majorization [9, 12, 10] which is able to monotonically reduce the value of the energy function. Although this type of optimization technique is not common in the vision community, it is very similar to Expectation Maximization (EM) type of algorithms.

4.1 Iterative Majorization

Iterative majorization is a monotonically convergent method developed in the area of statistics [9, 12, 10], this is able to solve relative complicated problems in a straight forward manner. The main idea is to find a function easier to minimize/maximize than the original one (e.g. quadratic function) at each iteration.

The first thing to do in order to minimize $G(\mathbf{B})$, eq. 4, is to find a function $L(\mathbf{B})$ which majorizes $G(\mathbf{B})$, that is, $L(\mathbf{B}) \geq G(\mathbf{B})$ and $L(\mathbf{B}_0) = G(\mathbf{B}_0)$, where $\mathbf{B}_0$ is the current estimate. The function $L(\mathbf{B})$ should be easier to minimize than $G(\mathbf{B})$. A minimum of $L(\mathbf{B})$, $\mathbf{B}_1$, is guaranteed to decrease the energy of $G(\mathbf{B})$. This is easy to show, since $L(\mathbf{B}_0) = G(\mathbf{B}_0) \geq L(\mathbf{B}_1) \geq G(\mathbf{B}_1)$. This is called the ”sandwich” inequality by De Leeuw [12]. Each update of the majorization will improve the value of the function, and if the function is bounded it will monotonically decrease the value of $L(\mathbf{B})$. Under these conditions it is always guaranteed to stop at a local optima.

Iterative majorization is very similar to EM [14] type of algorithms, which have been extensively used by the machine learning and computer vision communities. The EM algorithm is an iterative algorithm to find a local maxima of $log\ p(\mathbf{D}|\theta)$, where $\mathbf{D}$ is the data, θ are the parameters. Rather than maximizing directly the log likelihood, EM uses Jensen's inequality to find a lower bound $log\ p(\mathbf{D}|\theta) = log \int\ q(\mathbf{h})\frac{p(\mathbf{D},\mathbf{h}|\theta)}{q(\mathbf{h})}d\mathbf{h} \geq \int\ q(\mathbf{h})log\frac{p(\mathbf{D},\mathbf{h}|\theta)}{q(\mathbf{h})}d\mathbf{h}$, which holds for any distribution $q(\mathbf{h})$. The Expectation step, performs a functional approximation on this lower bound, that is, it finds the distribution $q(\mathbf{h})$ which maximizes the data and touches the log likelihood at the current parameter estimates θ_n. In fact, the optimal $q(\mathbf{h})$ is the posterior probability of the latent/hidden parameters given the data (i.e. $p(\mathbf{h}|\mathbf{D})$). In the maximization step, we maximize the lower bound w.r.t the parameters θ. The E-step in EM would be equivalent to the construction of the majorization function and the M-step just minimizes/maximizes this upper/lower bound.

4.2 Constructing a majorization function

In order to find a function which majorizes $G(\mathbf{B})$, we depart from the inequality [10], assuming the following factorization holds $\mathbf{A}_i = \mathbf{A}_i^{\frac{1}{2}}\mathbf{A}_i^{\frac{1}{2}}$ and $\mathbf{\Sigma}_i = \mathbf{\Sigma}_i^{\frac{1}{2}}\mathbf{\Sigma}_i^{\frac{1}{2}}$:

$$||(\mathbf{B}^T\mathbf{\Sigma}_i\mathbf{B})^{-\frac{1}{2}}\mathbf{B}^T\mathbf{A}_i^{\frac{1}{2}} - (\mathbf{B}^T\mathbf{\Sigma}_i\mathbf{B})^{\frac{1}{2}}(\mathbf{B}_n^T\mathbf{\Sigma}_i\mathbf{B}_n)\mathbf{B}_n^T\mathbf{A}_i^{\frac{1}{2}}||_F \geq 0$$

Rearranging this equation, it is easy to show the following inequality:

$$tr((\mathbf{B}^T\mathbf{\Sigma}_i\mathbf{B})^{-1}(\mathbf{B}^T\mathbf{A}_i\mathbf{B})) \geq 2tr((\mathbf{B}_n^T\mathbf{\Sigma}_i\mathbf{B}_n)^{-1})(\mathbf{B}_n^T\mathbf{A}_i\mathbf{B})) - tr\big((\mathbf{B}^T\mathbf{\Sigma}_i\mathbf{B})^{-1}(\mathbf{B}_n^T\mathbf{\Sigma}_i\mathbf{B}_n)^{-1}(\mathbf{B}_n^T\mathbf{A}_i\mathbf{B}_n)(\mathbf{B}_n^T\mathbf{\Sigma}_i\mathbf{B}_n)^{-1}\big)$$

Observe that just adding a sum to both sides of this inequality we obtain a function $L(\mathbf{B})$ which majorizes $G(\mathbf{B})$, that is:

$$\begin{aligned} G(\mathbf{B}) = -\textstyle\sum_i tr((\mathbf{B}^T\mathbf{\Sigma}_i\mathbf{B})^{-1}(\mathbf{B}^T\mathbf{A}_i\mathbf{B})) \leq L(\mathbf{B}) = -\textstyle\sum_i 2tr((\mathbf{B}_n^T\mathbf{\Sigma}_i\mathbf{B}_n)^{-1})(\mathbf{B}_n^T\mathbf{A}_i\mathbf{B})) \\ +tr\big((\mathbf{B}^T\mathbf{\Sigma}_i\mathbf{B})^{-1}(\mathbf{B}_n^T\mathbf{\Sigma}_i\mathbf{B}_n)^{-1}(\mathbf{B}_n^T\mathbf{A}_i\mathbf{B}_n)(\mathbf{B}_n^T\mathbf{\Sigma}_i\mathbf{B}_n)^{-1}\big) \end{aligned}$$

Effectively, it can easily shown that $L(\mathbf{B})$ majorizes $G(\mathbf{B})$ since $G(\mathbf{B}_n) = L(\mathbf{B}_n)$ and $L(\mathbf{B}) \geq G(\mathbf{B})$.

The function $L(\mathbf{B})$ is quadratic in $\mathbf{B}$ and hence easier to minimize. After rearranging terms a necessary condition for the minimum of $L(\mathbf{B})$ has to satisfy:

$$\begin{gathered} \tfrac{\partial L}{\partial \mathbf{B}} = \textstyle\sum_i -\mathbf{T}_i + \mathbf{\Sigma}_i\mathbf{B}\mathbf{F}_i = \mathbf{0} \\ \mathbf{F}_i = (\mathbf{B}_n^T\mathbf{\Sigma}_i\mathbf{B}_n)^{-1}(\mathbf{B}_n^T\mathbf{A}_i\mathbf{B}_n)(\mathbf{B}_n^T\mathbf{\Sigma}_i\mathbf{B}_n)^{-1} \quad \mathbf{T}_i = \mathbf{A}_i^T\mathbf{B}_n^T(\mathbf{B}_n^T\mathbf{\Sigma}_i\mathbf{B}_n)^{-1} \end{gathered} \tag{5}$$

solving eq. 5 involves solving the following system of linear equations $\sum_i \mathbf{T}_i = \sum_i \mathbf{\Sigma}_i\mathbf{B}\mathbf{F}_i$. The solution could be found in closed form by vectorizing eq. 5 with the help of Kronecker products. However the system would have dimensions of $(d \times l) \times (d \times l)$ which is not efficient in either space or time. Instead, we use an iterative algorithm which minimizes:

$$E(\mathbf{B}) = min_{\mathbf{B}}||\sum_i(\mathbf{T}_i - \mathbf{\Sigma}_i\mathbf{B}\mathbf{F}_i)||_F \tag{6}$$

Due to the huge number of the equations to solve ($d \times l$), an effective and linear time algorithm to solve for the optima is a normalized gradient descent:

$$\begin{gathered} \mathbf{B}^{n+1} = \mathbf{B}^n - \eta\tfrac{\partial E(\mathbf{B})}{\partial \mathbf{B}} \\ \mathbf{R}_k = \tfrac{\partial E(\mathbf{B})}{\partial \mathbf{B}} = -\textstyle\sum_i \mathbf{\Sigma}_i\mathbf{B}\mathbf{F}_i^T + \sum_i\sum_k \mathbf{\Sigma}_i^T\mathbf{\Sigma}_k\mathbf{B}\mathbf{F}_i\mathbf{F}_k^T \end{gathered} \tag{7}$$

η is the step size needed to converge. We find it by an optimization criteria which minimizes, $\eta = min_\eta||\sum_i \mathbf{T}_i - \sum_i \mathbf{\Sigma}_i(\mathbf{B} + \eta\mathbf{R}_k)\mathbf{F}_i)||$. After some derivation, it can be shown that $\eta = \frac{\sum_i\sum_k tr(\mathbf{\Sigma}_i\mathbf{R}_k\mathbf{T}_i\mathbf{T}_k^T\mathbf{B}^T\mathbf{\Sigma}_k) - \sum_i\left(\mathbf{\Sigma}_i\mathbf{R}_k\mathbf{T}_i\mathbf{B}^T\right)}{\sum_i\sum_k tr(\mathbf{\Sigma}_i\mathbf{R}_i\mathbf{T}_i\mathbf{T}_k^T\mathbf{R}_k^T\mathbf{\Sigma}_k)}$.

5 Dealing with high dimensional data

Learning discriminative models from high dimensional data such as images requires several strategies to get good generalization and computational tractability (e.g. feature selection or dimensionality reduction). In this context LDA or ODA can be a good initial step in order to preserve discriminative features in the low dimensional space. However, as it is well known dimensionality reduction techniques such as LDA, which preserve discriminative power can not handle very well the case that $n << d$ (more dimensions that training data), which is the typical one. For instance, an image of 100×100 pixels will correspond to feature vectors of 10000 dimensions, which will induce covariance matrices of 10000×10000. To make the covariance full rank, we would need at least 10000 independent samples available, even that would be a poor estimate.

In order to be able to generalize better than LDA and do not suffer from storage and computational requirements, we approximate the covariance matrices as the sum of outer products plus a diagonal matrix. We tried three of such factorizations:

$$\mathbf{\Sigma}_i \approx \mathbf{U}_i\Lambda_i\mathbf{U}_i^T + \sigma_i^2\mathbf{I}_d \quad \mathbf{\Sigma}_i \approx \mathbf{U}_i\Lambda_i\mathbf{U}_i^T + \beta_i^2(\mathbf{I}_d - \mathbf{U}_i\mathbf{U}_i^T) \quad \mathbf{\Sigma}_i \approx \mathbf{U}_i\Lambda_i\mathbf{U}_i^T + \Psi_i \tag{8}$$

where $\mathbf{U}_i \in \Re^{d\times l}$, $\Lambda_i \in \Re^{l\times l}$ is a diagonal matrix with the eigenvalues and $\Psi_i \in \Re^{d\times d}$ is a diagonal matrix. In order to estimate the parameters $\sigma_i^2,\ \beta_i^2,\ \mathbf{U}_i,\ \Lambda_i,\ \Psi_i$, a fitting approach is followed. For instance, $\mathbf{U}_i, \Lambda_i, \sigma_i^2$ are obtained by minimizing $E_c(\mathbf{U}_i, \Lambda_i, \sigma_i^2) = ||\mathbf{\Sigma}_i - \mathbf{U}_i\Lambda_i\mathbf{U}_i^T - \sigma_i^2\mathbf{I}_d||_F$. It can be shown that the optimal solution satisfies $\mathbf{U}_i\mathbf{\Sigma}_i = \mathbf{U}_i\hat{\Lambda}_i$, $\Lambda = \hat{\Lambda}_i - \sigma_i^2\mathbf{I}_d$ and $\sigma_i^2 = tr(\mathbf{R})/d$, where $\mathbf{R} = \mathbf{\Sigma}_i - \mathbf{U}_i\hat{\Lambda}_i\mathbf{U}_i^T$. In this case the eigenvectors to choose are the ones corresponding to biggest eigenvalues, however in the second factorization this would not necessarily has to be the case, see [18]. Finding the necessary conditions for the optima in the other two cases derives in $\beta_i^2 = tr(\mathbf{P}^T\mathbf{R})/tr(\mathbf{P}^T\mathbf{P}) = tr(\mathbf{R})/(d-l)$, where $\mathbf{P} = \mathbf{I} - \mathbf{U}_i\mathbf{U}_i^T$ and $\Psi = diag(\mathbf{R})/d$.

It is worth to point out two important aspects of the previous factorizations. Factorizing the covariance as the sum of outer products and a diagonal matrix is an efficient (in space and time) manner to deal with the small sample case, since we can compute $\mathbf{\Sigma}_i\mathbf{B} = \mathbf{U}_i\Lambda_i(\mathbf{U}_i^T\mathbf{B}) + \sigma_i^2\mathbf{B}$. In this case we do not need to explicitly have the full covariance matrix. On the other hand observe that the original covariance has $d(d+1)/2$ free parameters, and for instance in the first approximation of eq. 8 the number of parameters is reduced to $l(2d-l+1)/2$ (assuming orthonormality of $\mathbf{U}_i$), so we need much less data to estimate these parameters and hence it is not so prone to over-fitting.

6 Experiments

6.1 Toy Problem

In order to verify that under ideal conditions ODA outperforms LDA, we tested it on a toy problem. We have generated five 20-dimensional gaussian classes (d=20). Each sample from class c was generated by $\mathbf{y}_i = \mathbf{B}_c\mathbf{c} + \boldsymbol{\mu}_c + \mathbf{n}$, where $\mathbf{y}_i \in \Re^{20\times 1}$, $\mathbf{B}_c \in \Re^{20\times 7}$, $\mathbf{c} \sim N_7(\mathbf{0}, \mathbf{I})$ and $\mathbf{n} \sim N_{20}(\mathbf{0}, 3\mathbf{I})$. The means of each class are $\boldsymbol{\mu}_1 = 2\mathbf{1}_{20}$, $\boldsymbol{\mu}_2 = \mathbf{0}_{20}$, $\boldsymbol{\mu}_3 = -2[\mathbf{0}_{10}\ \mathbf{1}_{10}]^T$, $\boldsymbol{\mu}_4 = 2[\mathbf{1}_{10}\ \mathbf{0}_{10}]^T$, $\boldsymbol{\mu}_5 = 2[\mathbf{1}_5\ \mathbf{0}_5\ \mathbf{1}_5\ \mathbf{0}_5]^T$. The basis $\mathbf{B}_c$ are random matrices, where each element has been generated from $N(0,5)$. We impose a weak orthogonality between matrices (i.e. $tr(\mathbf{B}_i^T\mathbf{B}_j) = 0\ \forall i \neq j$), with a Gram-Schmidt approach, i.e. $\mathbf{B}_j = \mathbf{B}_j - \sum_{i=1}^{j-1} tr((\mathbf{B}_i\mathbf{B}_i)^{-1}\mathbf{B}_j^T\mathbf{B}_i)\mathbf{B}_i\ \forall j = 2\cdots 5$. We have generated 200 samples per class and we have approximated the covariance matrices as $\mathbf{\Sigma}_i = \mathbf{U}_i\mathbf{U}_i^T + \sigma_i^2\mathbf{I}$, such that they preserve 90% of the energy. In order to classify the test set, we use a linear classifier, that is, for a new sample $\mathbf{d}_i$, we project into the subspace by $\mathbf{x}_i = \mathbf{B}^T\mathbf{d}_i$ and we assign it to the class that has smallest distance, $(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_i)\hat{\mathbf{\Sigma}}_i^{-1}(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_i) + log|\hat{\mathbf{\Sigma}}_i|$, where $\hat{\boldsymbol{\mu}}_i$ and $\hat{\mathbf{\Sigma}}_i$ are the low dimensional estimates of the class covariance. Table 6.1 shows the average recognition rate of LDA and ODA over 50 trials. For each trial and each basis, we run the algorithm 5 times from different initial conditions (perturbing the LDA solution) and take the best solution. As we can observe from table 6.1 ODA always outperforms LDA and it is able to extract more features.

It is well known, that when $d >> n$ (small sample case), PCA can outperform LDA [13]. We run the same experiment as before but the dimension of each sample is 152 (i.e.

Basis	1	2	3	4	5	6	7
LDA	0.20	0.41	0.47	0.54	NA	NA	NA
ODA	0.2	0.60	0.72	0.81	0.88	0.92	0.95

Table 1: Average over 50 trials.

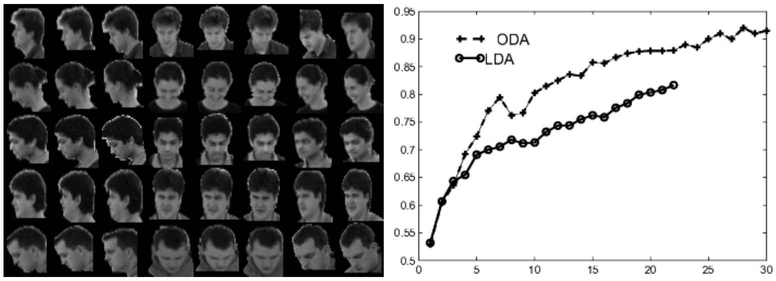


Figure 3: a) Taining images. b) LDA vs ODA

d=152) and just 40 samples per class are generated. The results can be seen in table 6.2.

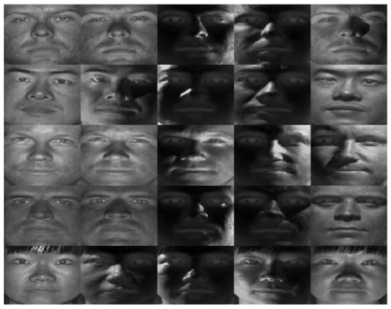
Basis	1	2	3	4	5	6
PCA	0.51	0.64	0.68	0.77	NA	NA
LDA	0.52	0.64	0.74	0.81	NA	NA
ODA	0.50	0.84	0.90	0.95	0.98	0.99
PCLDA	0.57	0.76	0.87	0.94	NA	NA
PCODA	0.47	0.83	0.90	0.95	0.97	0.99

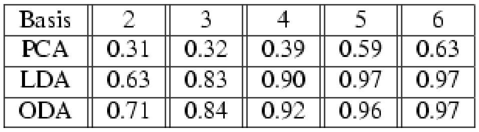
Table 2: Average over 50 trials.

PCLDA refers to do LDA onto the PCA projection (preserving 95% of the energy) and PCODA is the same with ODA. As we can see in the small sample case, ODA generally outperforms all the other methods. By projecting onto the principal components, we avoid the overfitting, and that is why PCLDA performs better than LDA. However, ODA does it implicitly and performs similarly to PCLDA but more features can be extracted.

6.2 Face Recognition

Face recognition is one of the classical pattern recognition problems which suffers from noise, limited number of training data and high dimensional spaces. In this experiment, we took the MOBO database [8], where people are walking on a treadmill and automatically segment (simple background subtraction) the heads of 24 people under 3 different poses. The head is in similar position, but it is not registered, the registration is explained in [4]. Figure 3 shows some samples of 5 people in the database. The images are 75×85 pixels and we have 240 samples for each class in the training and 240 samples in the testing. We project all the data into the ($240 \times 24 = 5760$) principal components, in practice we drop the eigenvectors corresponding to zero eigenvalues, practically we end up having around 4372 dimensional vectors and not the original 6375. In fig. 3.b the recognition rates for ODA vs. LDA are shown. We can see that ODA outperforms LDA. The errors are mostly due non perfect segmentation or registration.





Basis	2	3	4	5	6
PCA	0.31	0.32	0.39	0.59	0.63
LDA	0.63	0.83	0.90	0.97	0.97
ODA	0.71	0.84	0.92	0.96	0.97

Figure 4: a) Training data. b) Recognition rates.

We also tested ODA on the Yale face database B (http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html). We took the frontal pose of 10 subjects under 64 illumination conditions, and use 75% of the images for training and 25% for testing. The size of each image is 98×75 pixels. Fig. 4.a shows some images of the training data and fig. 4.b shows the recognition performance for PCA,LDA and ODA. In this case ODA performs slightly better than LDA due to the nature of the problem. As before, we have projected the data onto the principal components with non-zero eigenvalues.

7 Discussion and future work

In this paper we have introduced ODA which extends LDA taking into account the orientation of the classes. Additionally a linear time algorithm to search for a local optima is presented and a simple factorization method will be able to deal with the small size problem. Several synthetic and real experiments confirms that ODA outperforms classical LDA. However, several issues remains unsolved, there is need for optimization algorithms which improve the convergence and can find global optimal solutions. It could be interesting to train ODA using recent techniques of Adaboost, boosting, etc [6] which use a greedy strategy to look for local optima but they improve generalization. On the other hand, in the context of face recognition from video, one of the most important steps is registration and being able to deal with outliers and missing data.

References

[1] Canonical variate analysis - a general formulation. *Australian Journal of Statistics*, 26:86–96, 1984.

[2] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (19):711–720, 1997.

[3] L. Chen, H. Liao, M. Ko, J. Lin, and G. Yu. A new ldabased face recognition system which can solve the small sample size problem. *Pattern Recognition*, 33(10):1713–1726, 2000.

[4] F. de la Torre and T. Kanade. Multimodal oriented discriminant analysis. *In preparation for submission to PAMI*.

[5] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley and Sons Inc., 2001.

[6] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. Technical report, Dept. of Statistics, Stanford University Technical Report, 1998.

[7] K. Fukunaga. *Introduction to Statistical Pattern Recognition, Second Edition*. Academic Press.Boston, MA, 1990.

[8] R. Gross and J. Shi. The cmu motion of body (mobo) database. Technical Report TR-01-18, CMU-RI-, 2001.

[9] J.W. Heiser. *Convergent computation by iterative majorization; theory and applications in multidimensional data analysis.* Krzanowski ed., Oxford Univ. Press, 1997.

[10] Henk A. L. Kiers. Maximization of sums of quotients of quadratic forms and some generalizations. *Psychometrika*, 60(2):221–245, 1995.

[11] N. Kumar and A. Andreou. Heteroscedastic discriminant analysis and reduced rank hmms for improved speech recognition. *Speech Communication*, 26(4), 1998.

[12] J. De Leeuw. *Block relaxation algorihtms in statistics*. H.H. Bock, W. Lenski, M. Ritcher eds. Information Systems and Data Analysis. Springer-Verlag., 1994.

[13] A.M. Martinez and A.C. Kak. Pca versus lda. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2003.

[14] R. Neal and G. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. M. I. Jordan, ed, *Learning in Graphical Models*. Kluwer, 1998.

[15] G. Saon, M. Padmanabhan, R. Gopinath, and S. Chen. Maximum likelihood discriminant feature spaces. In *ICASSP*, 2000.

[16] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. Jorunal of Computer Vision.*, 9(2):137–154, 1992.

[17] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal Cognitive Neuroscience*, 3(1):71–86, 1991.

[18] M. Welling, F. Agakov, and C. Williams. Extreme components analysis. *NIPS*, 2003.

[19] S. Gong Y. Li and H. Liddell. Recognising trajectories of facial identities using kernel discriminant analysis. *Image and Vision Computing*, 21(13-14), 2003.

[20] H. Yu and J. Yang. A direct lda algorith for high-dimensional data– with applicatins to face recognition. *Pattern Recognition*, 34(10):2067–2070, 2001.

[21] W. Zhao. Discriminant component analysis for face recognition. In *ICPR*, pages 818–821, 2000.