Probabilistic Linear Discriminant Analysis

Sergey Ioffe*

Fujifilm Software, 1740 Technology Dr., Ste. 490, San Jose, CA 95110
sioffe@gmail.com

Abstract. Linear dimensionality reduction methods, such as LDA, are often used in object recognition for feature extraction, but do not address the problem of how to use these features for recognition. In this paper, we propose Probabilistic LDA, a generative probability model with which we can both extract the features and combine them for recognition. The latent variables of PLDA represent both the class of the object and the view of the object within a class. By making examples of the same class share the class variable, we show how to train PLDA and use it for recognition on previously unseen classes. The usual LDA features are derived as a result of training PLDA, but in addition have a probability model attached to them, which automatically gives more weight to the more discriminative features. With PLDA, we can build a model of a previously unseen class from a single example, and can combine multiple examples for a better representation of the class. We show applications to classification, hypothesis testing, class inference, and clustering, on classes not observed during training.

1 Introduction

There is a long tradition of using linear dimensionality reduction methods for object recognition [1, 2]. Most notably, these include Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). While PCA identifies the linear subspace in which most of the data's energy is concentrated, LDA identifies the subspace in which the data between different classes is most spread out, relative to the spread within each class. This makes LDA suitable for recognition problems such as classification. One of the questions that dimensionality reduction methods do not answer is: what do we do with the lower-dimension representation of the data? A common technique is to project the data onto a PCA subspace, thus eliminating singularities, and then find an LDA subspace. However, after the projection, how do we combine the components of the resulting multivariate representation? Clearly some dimensions (for example, the dominant projection directions identified by LDA) have to be more important than others, but how do we incorporate this difference in importance into recognition? How do we perform tasks such as classification and hypothesis testing on examples of classes we haven't seen before, and how do we take advantage of multiple examples of a new class?

In this paper, we reformulate the problem of dimensionality reduction for recognition in the probabilistic context. It has long been known that LDA maximizes the likelihood of a Gaussian mixture model and is mathematically equivalent to linear regression of

* The author is currently at Google.

A. Leonardis, H. Bischof, and A. Pinz (Eds.): ECCV 2006, Part IV, LNCS 3954, pp. 531–542, 2006.

the class assignment labels [3, 4]. Such regression, however, is useful only when LDA is used to classify examples of the classes represented in the training data. One of the many problems in which this assumption is false is face recognition. For example, having trained a system, we need to be able to determine whether two face views belong to the same person, even though we have not seen this person before. In these cases, we are not able to build an accurate probability model for the new person (since we have only one example), nor is a discrete class label defined for an example of a previously unseen class.

In a Gaussian mixture model with common class-conditional covariances, each class is described by its center, and the support of the prior distribution of the class centers is a finite set of points. This is not sufficient for handling new classes, and in this work we solve this problem by making the prior of the class centers continuous. We can learn this prior (which models the differences between classes) as well as the common variance of the class-conditional distributions (which models the differences between examples of the same class). We will show that by maximizing the model likelihood we arrive at the features obtained by Linear Discriminant Analysis. However, in Probabilistic LDA, we also obtain a principled method of combining different features so that the more discriminative features have more impact on recognition.

Probabilistic LDA is a general method that can accomplish a wide variety of recognition tasks. In "one-shot learning" [5], a single example of a previously unseen class can be used to build the model of the class. Multiple examples can be combined to obtain a better representation of the class. In hypothesis testing, we can compare two examples, or two groups of examples, to determine whether they belong to the same (previously unseen) class. This can further be used to cluster examples of classes not observed before, and automatically determine the number of clusters.

The method proposed in this paper is to LDA what Probabilistic PCA [6] is to PCA. Namely, we will derive the commonly used feature extraction method using a probabilistic approach, and obtain the method not just to compute the features, but also to combine them. While PPCA is used to model a probability density of data, PLDA can be used to make probabilistic inferences about the class of data.

2 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is commonly used to identify the linear features that maximize the between-class separation of data, while minimizing the within-class scatter [7]. Consider a training data set containing N examples $\{\mathbf{x}^1 \dots \mathbf{x}^N\}$, where each example $\mathbf{x}^i$ is a column vector of length d. Each training example belongs to one of the K classes. Let $\mathcal{C}_k$ be the set of all examples of class k, and let $n_k = |\mathcal{C}_k|$ be the number of examples in class $k = 1 \dots K$. In LDA, the within-class and between-class scatter matrices are computed:

$$S_w = \frac{\sum_k \sum_{i \in \mathcal{C}_k} (\mathbf{x}^i - \mathbf{m}_k)(\mathbf{x}^i - \mathbf{m}_k)^T}{N},\ S_b = \frac{\sum_k n_k (\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^T}{N} \quad (1)$$

where $\mathbf{m}_k = \frac{1}{n_k} \sum_{i \in \mathcal{C}_k} \mathbf{x}^i$ is the mean of kth class, and $\mathbf{m} = \frac{1}{N} \sum_i \mathbf{x}^i$ is the mean of the data set. We seek the linear transformation $\mathbf{x} \rightarrow W^T \mathbf{x}$ that maximizes the

between-class variance relative to the within-class variance, where W is a $d \times d'$ matrix, with d' being the desired number of dimensions. It can be shown that the columns of the optimal W are the generalized eigenvectors such that $S_b \mathbf{w} = \lambda S_w \mathbf{w}$, corresponding to the d' largest eigenvalues. One consequence of this result is that W simultaneously diagonalizes the scatter matrices $W^T S_b W$ and $W^T S_w W$. In other words, LDA decorrelates the data both between and within classes.

The LDA projections can be derived by fitting a Gaussian Mixture Model to the training data [3]. The mixture model that results can be used to classify examples of the classes represented in the training data, but not the novel classes. A different probability model is required for that purpose, and is provided by Probabilistic LDA.

3 Probabilistic LDA

A Gaussian mixture model can be thought of as a latent variable model where the observed node $\mathbf{x}$ represents the example, and the latent variable $\mathbf{y}$ is the center of a mixture component and represents the class (Fig. 1a). Members of the same class share the class variable $\mathbf{y}$. The class-conditional distributions

$$P(\mathbf{x} \,|\, \mathbf{y}) = \mathcal{N}(\mathbf{x} \,|\, \mathbf{y}, \Phi_w)$$

have a common covariance matrix Φ_w, and the prior on the class variable assigns a probability mass to each of the finite number of points: $P(\mathbf{y}) = \sum_{k=1}^{K} \pi_k \delta(\mathbf{y} - \mu_k)$. When the centers μ_k are constrained to lie in a low-dimensional (but unknown) subspace, likelihood maximization with respect to μ_k, π_k and Φ_w recovers the standard LDA projections [3]. We want to extend the probabilistic framework to be able to handle classes not represented in the training data. To this end, we propose to modify the latent variable prior and make it continuous. In particular, to enable efficient inference and closed-form training, we shall impose a Gaussian prior:

$$P(\mathbf{y}) = \mathcal{N}(\mathbf{y} \,|\, \mathbf{m}, \Phi_b)$$

We will require Φ_w to be positive definite, and Φ_b to be positive semi-definite. It is a well-known result from linear algebra that Φ_w and Φ_b can be simultaneously diagonalized: $V^T \Phi_b V = \Psi$ and $V^T \Phi_w V = \mathbf{I}$, where the diagonal matrix Ψ and non-singular matrix V are found by solving a generalized eigenproblem. By defining $A = V^{-T}$, we have $\Phi_w = AA^T$ and $\Phi_b = A\Psi A^T$. Our model is then:

$$\boxed{\begin{array}{ll} \mathbf{x} = \mathbf{m} + A\mathbf{u} & \text{where} \\ \mathbf{u} \sim \mathcal{N}(\cdot \,|\, \mathbf{v}, \mathbf{I}) & \text{and} \\ \mathbf{v} \sim \mathcal{N}(\cdot \,|\, 0, \Psi) & \end{array}} \tag{2}$$

Here $\mathbf{v}$ represents the class, and $\mathbf{u}$ represents an example of that class in the projected space — just as $\mathbf{y} = \mathbf{m} + A\mathbf{v}$ and $\mathbf{x} = \mathbf{m} + A\mathbf{u}$ do in the data space. Here, Ψ is diagonal, $\Psi \geq 0$. The corresponding graphical model is shown in Fig. 1b.

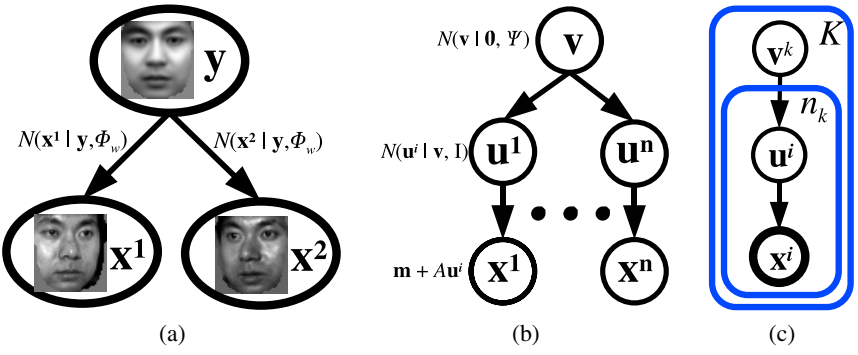


Fig. 1. (a) Modeling class and view. The latent variable $\mathbf{y}$ represents the class center, and the examples of the class are drawn from a Gaussian distribution centered at $\mathbf{y}$. If the prior on $\mathbf{y}$ is discrete, this is a mixture model. For the model to generalize to previously unseen classes, we instead impose a Gaussian prior $\mathcal{N}(\mathbf{y} \mid \mathbf{m}, \Phi_b)$ on the class center, which leads to Probabilistic LDA. **(b)** By diagonalizing the covariances Φ_b and Φ_w, PLDA models the class center $\mathbf{v}$ and examples $\mathbf{u}$ in the latent space where the variables are independent. The example $\mathbf{x}$ in the original space is related to its latent representation $\mathbf{u}$ via an invertible transformation A. All the recognition activities take place in the latent space. **(c)** A set of examples $\mathbf{x}$ grouped into K clusters, where examples within the kth cluster share the class variable $\mathbf{v}^k$. The latent variables $\mathbf{v}$ and $\mathbf{u}$ are hidden and can be integrated out. In the training data, the grouping of examples into clusters is given, and we learn the model parameters by maximizing the likelihood. If, instead, the model parameters are fixed, likelihood maximization with respect to the class assignment labels solves a clustering problem.

3.1 Inference in the Latent Space

The main advantage of PLDA is that it allows us to make inference about the classes not present during training. One example of such a situation is face recognition. The model parameters are learned from training data, but the trained system must deal with examples of novel individuals. This is different from many other object recognition tasks where the training data contains examples of all the classes.

In the problem of classification, we are given a gallery $(\mathbf{x}^1 \dots \mathbf{x}^M)$ containing one example from each of M classes, as well as a probe example $\mathbf{x}^p$. We know that the probe belongs to one of the M classes in the gallery, and need to determine which one. We will answer this question by maximizing the likelihood. This is more easily accomplished in the latent space, where we apply the transform $\mathbf{u} = A^{-1}(\mathbf{x} - \mathbf{m})$ to all of the data, which decorrelates the data as shown in Eqn. (2). Consider an example $\mathbf{u}^g$ from the gallery. Let us compute $P(\mathbf{u}^p \mid \mathbf{u}^g)$, the probability of the probe example coming from the same class as the gallery example. By performing the inference on the class variable, we have

$$P(\mathbf{v} \mid \mathbf{u}) = \mathcal{N}(\mathbf{v} \mid \tfrac{\Psi}{\Psi+\mathbf{I}}\mathbf{u}, \tfrac{\Psi}{\Psi+\mathbf{I}}) \tag{3}$$

Since $\mathbf{u}^p$ and $\mathbf{u}^g$ are conditionally independent given $\mathbf{v}$ (see Fig. 1), we have

$$P(\mathbf{u}^p \mid \mathbf{u}^g) = \mathcal{N}(\mathbf{u}^p \mid \tfrac{\Psi}{\Psi+\mathbf{I}}\mathbf{u}^g, \mathbf{I} + \tfrac{\Psi}{\Psi+\mathbf{I}}) \tag{4}$$

To classify a probe example, we compute $P(\mathbf{u}^p \mid \mathbf{u}^g)$ for $g = 1 \ldots M$, and pick the maximum. With PLDA, we were able to combine the knowledge about the general structure of the data, obtained during training, and the examples of new classes, yielding a principled way to perform classification[1].

We can also combine multiple examples of a class into a single model, improving the recognition performance. If n independent examples $\mathbf{u}^g_{1\ldots n}$ of a class are in the gallery to be used for classification, then we can show that

$$P(\mathbf{u}^p \mid \mathbf{u}^g_{1\ldots n}) = \mathcal{N}(\mathbf{u}^p \mid \tfrac{n\Psi}{n\Psi+\mathbf{I}}\bar{\mathbf{u}}^g, \mathbf{I} + \tfrac{\Psi}{n\Psi+\mathbf{I}})$$

where $\bar{\mathbf{u}}^g = \frac{1}{n}(\mathbf{u}^g_1 + \cdots + \mathbf{u}^g_n)$.

Another common recognition problem is that of hypothesis testing. Given two examples of previously unseen classes, we need to determine whether they belong to the same class. Methods such as LDA do not solve this problem, but with PLDA it is easily accomplished. For two examples $\mathbf{u}^p$ and $\mathbf{u}^g$, we compute the likelihoods $P(\mathbf{u}^p)P(\mathbf{u}^g)$ and $\mathcal{P}(\mathbf{u}^p, \mathbf{u}^g) = \int P(\mathbf{u}^p \mid \mathbf{v})P(\mathbf{u}^g \mid \mathbf{v})P(\mathbf{v})d\mathbf{v}$ corresponding to the two examples belonging to different classes and the same class, respectively, and use the ratio of the two to classify. More generally, if the probe contains multiple examples of an object and the gallery contains multiple examples of another object, we compute the likelihood ratio

$$R(\{\mathbf{u}^p_{1\ldots m}\}, \{\mathbf{u}^g_{1\ldots n}\}) = \frac{\text{likelihood(same)}}{\text{likelihood(diff)}} = \frac{\mathcal{P}(\mathbf{u}^p_{1\ldots m}, \mathbf{u}^g_{1\ldots n})}{\mathcal{P}(\mathbf{u}^p_{1\ldots m})\mathcal{P}(\mathbf{u}^g_{1\ldots n})} \tag{5}$$

where

$$\begin{aligned}\mathcal{P}(\mathbf{u}^{1\ldots n}) &= \int P(\mathbf{u}^1 \mid \mathbf{v}) \cdots P(\mathbf{u}^n \mid \mathbf{v})P(\mathbf{v})d\mathbf{v} \\ &= \textstyle\prod_{t=1}^{d} \frac{1}{(2\pi)^{n/2}(\psi_t + \frac{1}{n})^{1/2}} \exp\left(-\frac{(\bar{u}_t)^2}{2(\psi_t + \frac{1}{n})} - \frac{\sum_{i=1}^{n}(u_t^i - \bar{u}_t)^2}{2}\right)\end{aligned} \tag{6}$$

is the distribution of a set of examples, given that they belong to the same class. Here, for the tth feature, $\bar{u}_t = \frac{1}{n}\sum_{i=1}^{n} u_t^i$. Since Ψ is diagonal, the contributions of different features to $\mathcal{P}$ are decoupled. For priors π_{same} and π_{diff}, the probability that all the examples are of the same class is $(1 + \frac{\pi_{\text{diff}}/\pi_{\text{same}}}{R})^{-1}$. If $R > \frac{\pi_{\text{diff}}}{\pi_{\text{same}}}$, the two groups of examples belong to the same class; otherwise, they do not. Being able to compare two groups of examples makes it also possible to use PLDA for clustering.

The between-class feature variances ψ_t indicate how discriminative the features are. In PLDA, the better features automatically contribute more to recognition. As a special case, consider a completely non-discriminative feature, for which $\psi = 0$. It can be seen that this feature does not contribute to R (Eqn. (5)), or to the other equations above, at all. Therefore, we can perform dimensionality reduction by keeping only the rows of A^{-1} corresponding to non-zero ψ. If we want to use at most d' dimensions, we impose the constraint that no more than d' entries of Ψ be non-zero. We will show how to do this in the next section.

[1] The problem of outliers, not belonging to any of the gallery classes, is also solved by PLDA, where we define $P(\mathbf{u}^p \mid \emptyset) = \mathcal{N}(\mathbf{u}^p \mid 0, \Psi + \mathbf{I})$.

3.2 Learning the Model Parameters

The unknown parameters of PLDA are the mean $\mathbf{m}$, the covariance matrix Ψ, and the loading matrix A (or, equivalently, the variances Φ_b and Φ_w). These parameters can be learned in the maximum likelihood framework. Given N training patterns separated into K classes (Fig. 1c), we can compute the likelihood of the data. We will make the assumption that all examples are independently drawn from their respective classes. The log-likelihood is

$$\ell(\mathbf{x}^{1\dots N}) = \sum_{k=1}^{K} \ln \mathcal{P}(\mathbf{x}^i : i \in \mathcal{C}_k) \tag{7}$$

where

$$\mathcal{P}(\mathbf{x}^1 \dots \mathbf{x}^n) = \int \mathcal{N}(\mathbf{y} \,|\, 0, \Phi_b)\mathcal{N}(\mathbf{x}^1 \,|\, \mathbf{y}, \Phi_w) \cdots \mathcal{N}(\mathbf{x}^n \,|\, \mathbf{y}, \Phi_w) d\mathbf{y}$$

is the joint probability distribution of a set of n patterns, provided they belong to the same class. Computing the integral, we get: $\ln \mathcal{P}(\mathbf{x}^{1\dots n}) = C - \frac{1}{2}(\ln|\Phi_b + \frac{\Phi_w}{n}| + \mathrm{tr}((\Phi_b + \frac{\Phi_w}{n})^{-1}(\bar{\mathbf{x}} - \mathbf{m})(\bar{\mathbf{x}} - \mathbf{m})^T) + (n-1)\ln|\Phi_w| + \mathrm{tr}(\Phi_w^{-1}(\sum_{i=1}^{n}(\mathbf{x}^i - \bar{\mathbf{x}})(\mathbf{x}^i - \bar{\mathbf{x}})^T)))$ where $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}^i$ and C is a constant term that we can ignore.

Let us consider the case where each of the classes in the training data is represented by the same number n of examples. Maximizing Eqn. (7) with respect to $\mathbf{m}$, we find $\mathbf{m} = \frac{1}{N}\sum_i \mathbf{x}^i$. Substituting it back, we finally obtain

$$\begin{aligned}\ell(\mathbf{x}^{1\dots N}) = -\frac{c}{2}\big(\ & \ln|\Phi_b + \frac{1}{n}\Phi_w| + \mathrm{tr}((\Phi_b + \frac{1}{n}\Phi_w)^{-1}S_b) \\ & + (n-1)\ln|\Phi_w| + n\mathrm{tr}(\Phi_w^{-1}S_w)\big)\end{aligned} \tag{8}$$

where S_b and S_w are defined in Eqn. (1). We need to maximize the value of ℓ with respect to Φ_b and Φ_w, subject to Φ_w being positive definite, Φ_b being positive semi-definite, and, in the case of dimensionality reduction, $\mathrm{rank}(\Phi_b) \leq d'$. Without these constraints, simple matrix calculus would yield

$$\Phi_w = \tfrac{n}{n-1}S_w, \quad \Phi_b = S_b - \tfrac{1}{n-1}S_w$$

Therefore, if the scatter matrices S_w and S_b are diagonal then so are the covariances Φ_w and Φ_b. In fact, this diagonalization property holds even if the above constraints are imposed. According to Eqn. (2), $\Phi_b = A\Psi A^T$, where A is invertible. For fixed Ψ, unconstrained optimization of Eqn. (8) with respect to A^{-1} makes both $A^{-1}S_bA^{-T}$ and $A^{-1}S_wA^{-T}$ diagonal. Therefore, the columns of A^{-T} contain the generalized vectors of S_b and S_w, and the projection of data into the latent space (where the recognition takes place) is the LDA projection discussed in §2. Finally optimizing (8) with respect to Ψ, subject to $\Psi \geq 0$ and $\mathrm{rank}(\Psi) \leq d'$, we obtain the method for learning the parameters of our model (2). This method is shown in Fig. 2.

Our method was derived for the case where each class in the training data is represented by the same number n of examples. This may not be true in practice, in which case we can resample the data to make the number of examples the same, use EM (as shown in §5), or use approximations. We took the latter approach, using the closed-form solution in Fig. 2 where n was taken to be the average number of examples per class.

Given: Training examples $\mathbf{x}^{1\cdots N}$ from K classes, with $n = N/K$ examples per class
Find: Parameters $\mathbf{m}, A, \Psi$ maximizing the likelihood of the PLDA model (Eqn. (2), Fig. 1).

1. Compute the scatter matrices S_b and S_w (Eqn. (1)). Find the matrix W of generalized eigenvectors with columns such that $S_b\mathbf{w} = \lambda S_w\mathbf{w}$. Then, $\mathbf{x} \to W^T\mathbf{x}$ is the LDA projection, and $\Lambda_b = W^T S_b W$ and $\Lambda_w = W^T S_w W$ are both diagonal.
2. Set
$$\mathbf{m} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}^i$$
$$A = W^{-T}\left(\frac{n}{n-1}\Lambda_w\right)^{1/2}$$
$$\Psi = \max\left(0, \frac{n-1}{n}(\Lambda_b/\Lambda_w) - \frac{1}{n}\right)$$
3. To reduce the dimensionality to d', keep the d' largest elements of Ψ and set the rest to zero. In the latent space $\mathbf{u} = A^{-1}(\mathbf{x} - \mathbf{m})$, only the features corresponding to non-zero entries of Ψ are needed for recognition.

Fig. 2. Fitting the parameters of the PLDA model

4 Results

With Probabilistic LDA, we model the variations in the appearance of any object, as well as the differences in the appearance of different objects. This makes PLDA a general model, useful for a variety of recognition tasks on examples of previously unseen classes. We will show its applications to class inference, classification, hypothesis testing, and clustering.

4.1 Class Inference

By modeling both within-class and between-class variations, PLDA allows us to isolate the class component of an example. This emphasizes the features that make different objects distinct, discarding the information not useful for recognition.

From Eqn. (3), we can show that the MAP estimate (and also the expectation) of the class center $\mathbf{y}$ corresponding to example $\mathbf{x}$ is $\hat{\mathbf{y}} = \mathbf{m} + A\hat{\mathbf{v}} = \mathbf{m} + A(\Psi + \mathbf{I})^{-1}\Psi A^{-1}(\mathbf{x} - \mathbf{m})$. In Fig. 3, we demonstrate the class inference on faces from the PIE database [8]. Each row of Fig. 3a contains one person, but the view variations within each row are large. In Fig. 3b we show the estimate of the class center. Most of the variation within rows has been eliminated, while different rows look distinct.

4.2 Classification

One natural task for PLDA is classification, and we apply it to face recognition. We trained the system on a set faces extracted from videos, each of which was automatically cropped and contrast-normalized. We reduce the dimensionality using PCA and capturing around 96% of the energy. In the resulting subspace, we train the PLDA model as described in §3.

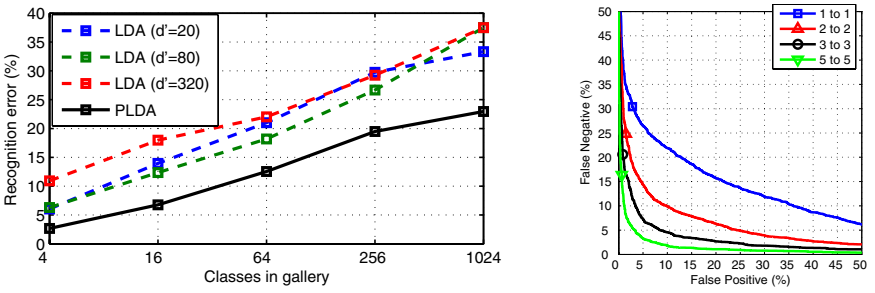
(a): example $\mathbf{x} = \mathbf{m} + A\mathbf{u}$ (b): class estimate $\hat{\mathbf{y}} = \mathbf{m} + A\hat{\mathbf{v}}$

Fig. 3. Class inference with PLDA. **(a)** Faces from the PIE dataset. Rows correspond to different people. **(b)** We estimate the class variable $\mathbf{y}$ from each example $\mathbf{x}$. This emphasizes the information relevant to recognition, and largely takes out the view variations. This makes the images within the same class look similar, and those of different classes different. The inference was done on each image independently. The system has never seen images from these classes before.

Each test case consists of a gallery containing one example of each of M people from the FERET database [9] (the training data was collected by us and did not include any FERET images). The probe $\mathbf{x}^p$ contains a different image of one of those M people, and is classified by maximizing the likelihood $P(\mathbf{x}^p \mid \mathbf{x}^g)$ (Eqn. (4)). In Fig. 4a we compare the performance of PLDA to that of LDA. In LDA-based classification, we project the data onto a d'-dimensional space, normalize it so that each feature has the same within-class variance, and classify the probe by finding the nearest neighbor from the gallery (equivalent to a maximum-likelihood decision rule). Although the features extracted by PLDA are the same as LDA, the probability model in PLDA makes it consistently outperform LDA of any dimensionality d', for any gallery size. Note that with PLDA we do not need to choose the best value for d', since the probability model automatically gives less importance to the less discriminative features. On the other hand, d' affects the performance of LDA (here, $d' = 80$ seems to be the best choice).

4.3 Hypothesis Testing

While PLDA lets us perform classification better than LDA, there are many tasks that LDA does not address at all. In hypothesis testing, we need to determine whether two examples belong to the same class or not. More generally, given two groups of examples, where each group belongs to one class, we need to determine whether the two classes are the same. This is accomplished by PLDA by comparing the likelihood ratio R (Eqn. (5)) with the prior ratio. We use the COIL database [10], containing 72 images of each of 100 objects. We randomly select 68 objects to use for training, and test on the 32 remaining objects. An error results when two examples of the same object selected from the test set are classified as different (false negative), or when two examples of different objects are classified as the same (false positive). The images were sampled to 32×32 pixels, and PCA (computed on the training set) was used to extract 200 features.



(a): classification (b): hypothesis testing

Fig. 4. (a) Evaluating the classification performance of LDA (with varying dimensions d') and PLDA on the FERET face data set. A test gallery contains M classes, with one example per class. The probe is a different example of one of the M classes, and needs to be labeled. We plot the misclassification rate as a function of M. PLDA significantly outperforms LDA. The training and test data came from different sources and have no people in common. **(b)** Hypothesis testing using PLDA. We determine whether two examples belong to the same class or not by comparing the likelihood ratio R with the prior ratio. The top curve shows the false positive and false negative rates computed for the COIL database, with the marker corresponding to equal priors. We can also compare two *groups* of examples, where each contains several examples of one class. Combining multiple examples yields better models of the new classes, reducing the error rates. Different classes were used for training and testing.

In Fig. 4b, we show the error rates, where the ratio of priors $\frac{\pi_{\text{diff}}}{\pi_{\text{same}}}$ moves us along the curve (the marker corresponds to equal priors). With PLDA we can compare groups of examples too, and we show that by comparing several examples of one class with several examples of the other we get much better accuracy than with single examples. We expect that a non-linear dimensionality reduction such as LLE [11] would make the data better suited for the Gaussian model in PLDA, further reducing the error rates.

4.4 Clustering

While in classification we have the gallery of labeled objects, a different, unsupervised approach is needed when no class labels are available. In that case, we need to cluster the examples, so that each cluster roughly corresponds to one class. Methods such as K-means can be used, but suffer from the arbitrary choice of metric and the need to specify the number of clusters in advance. With PLDA, we can automatically determine the optimal number of classes.

We approach clustering as the likelihood maximization problem. Each split of examples into clusters corresponds to a graphical model (Fig. 1c) in which all examples within one cluster share the class variable, and the likelihood of the clustering is computed by integrating out the class variables, which can be done in closed form (Eqn. (6)). Because the set of examples can be split into clusters in an exponential number of ways, we cannot compute the likelihood of each clustering. Instead, we use agglomerative clustering as an approximate search mechanism. We start with each example in its own cluster, and at each iteration merge two clusters. When two clusters are merged, the log-likelihood ℓ increases by $\ln R$, where R is the likelihood ratio defined in Eqn. (5).

Fig. 5. PLDA makes it possible to cluster examples and automatically determine the optimal number of clusters. We approach clustering as likelihood maximization, and use agglomerative clustering. At each step we merge the clusters with the largest likelihood ratio R; this increases the log-likelihood by $\ln R$. **(a)** The log-likelihood ℓ as a function of the number of clusters. The maximum is reached at 14 clusters. **(b)** The clusters maximizing the likelihood. If we give each person a label *A* through *H*, the clusters are: *(BBBBBDC), (AAAAA), (FFFFF), (DDDD), (IIII), (HHHH), (GGGG), (EEEE), (EG), (HC), (CC), (I), (C), (C).*

Therefore, at each iteration, we merge the two clusters with the maximum R, and update the log-likelihood as $\ell \leftarrow \ell + \ln R$. The point in this process at which ℓ reaches its maximum tells us the (approximately) optimal way to cluster the data, including the number of clusters.

We tested the clustering algorithm on the PIE dataset, by randomly selecting 5 images of each of the 9 dataset collectors (the training data didn't include any PIE images). In Fig. 5a we plot the log-likelihood ℓ against the number of clusters. The graph has a maximum, which tells us how many clusters are needed (14 in this case). Fig. 5b shows the corresponding clusters. While the clustering is not perfect, it largely corresponds to the true classes of the examples.

5 Combining Probabilistic PCA and Probabilistic LDA

Usually, a dimensionality reduction such as PCA must be used before applying LDA to eliminate singularities in the problem. Using PCA before PLDA works very well for recognition, but it may be desirable to use PLDA to model the probability distribution in the original space, and not the PCA-projected subspace. This suggests combining PLDA with Probabilistic PCA [6] instead.

Probabilistic PCA fits the data with a model $\mathbf{x} \sim \mathcal{N}(\cdot \,|\, \mathbf{m} + A\mathbf{u}, \Sigma)$ where the latent variable $\mathbf{u} \sim \mathcal{N}(\cdot \,|\, 0, \mathbf{I})$, and $\Sigma = \sigma^2\mathbf{I}$. We will combine PPCA with PLDA (Eqn. (2)), to obtain the following model:

$$\mathbf{x} \sim \mathcal{N}(\cdot \,|\, \mathbf{m} + A\mathbf{u}, \Sigma), \quad \text{where } \mathbf{u} \sim \mathcal{N}(\cdot \,|\, \mathbf{v}, \mathbf{I}) \quad \text{and } \mathbf{v} \sim \mathcal{N}(\cdot \,|\, 0, \Psi) \tag{9}$$

If D is the dimensionality of the data and d is the desired dimensionality of the latent space, we constrain A to be of size $D \times d$. We find the parameters of the model by using Expectation Maximization (e.g. [7]). Note that by letting $d = D$ and setting $\sigma \to 0$ we obtain an EM method for fitting the PLDA model which doesn't require that each class be represented by the same number of training examples.

We can further extend PPCA+PLDA to model wider, non-linear view variations, by defining a mixture model in which each mixture component j has its own linear transformation $(\mathbf{m}_j, A_j)$. We can think of A_j as coarsely representing the view, and $\mathbf{u} - \mathbf{v}$ as capturing finer view variations. The class variable $\mathbf{v}$ is shared by all examples of the same class, even those from different mixture components. The recognition tasks and EM-based training can be performed approximately, using an additional step assigning each example to one of the mixture components. This allows us to project each example into the latent space, and perform the recognition activities there. Note that if an example comes from a class represented by $\mathbf{v}$, and belongs to the jth mixture component, then its expected value is $\mathbf{m}_j + A_j\mathbf{v}$, which is the representation used in asymmetric bilinear models [12]. However, unlike the bilinear models, ours is a probability model, and training it does not require the ground-truth view labels, which may be hard to obtain. Experiments with the PPCA+PLDA mixture model will be a part of our future research.

6 Discussion

We presented a novel generative model that decomposes a pattern into the class and the view. Probabilistic Linear Discriminant Analysis (PLDA) is related to LDA and Probabilistic PCA, and can be thought of as LDA with a probability distributions attached to the features. The probability distribution models the data through the latent variables corresponding to the class and the view. This allows us to perform inference and recognition. The model automatically gives more importance to the more discriminative features, which helps us avoid a search for the optimal number of features. On the other hand, we can perform dimensionality reduction with PLDA, by imposing an upper limit on the rank of the between-class variance. As an extension, we also proposed a PPCA+PLDA model that doesn't require PCA pre-processing, and a PPCA+PLDA Mixture for modeling wider view variations.

One of the most important advantages of PLDA, compared to LDA and its previously proposed probabilistic motivations, is that the probability distributions are learned not only for the examples within a class but for the class center as well. This makes PLDA perfectly suited for a wide variety of recognition problems on classes we have not seen before. A model of a class can be built from a single example (one-shot learning), and is further improved by combining multiple examples of a class. We can perform classification ("what is the class of the example?"), hypothesis testing ("do the two examples belong to the same class?"), and clustering.

Just like any linear model, PLDA performs best when the data obey the linear assumptions. However, it can be applied to non-linear distributions if the features are extracted first that linearize the data. One option is to embed the data in a linear manifold (e.g. [11]), and use PLDA there. Alternatively, we can use the kernel trick inside PLDA, by extracting non-linear features from the data using Kernel LDA [13], and then computing the probability distribution of each feature independently.

Acknowledgments. Many thanks to David Forsyth, Thomas Leung and Troy Chinen for discussions and suggestions, and to the paper's area chair and reviewers for very helpful comments and literature pointers.

References

1. Belhumeur, P.N., Hespanha, J., Kriegman, D.J.: Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Trans. PAMI **19**(7) (1997) 711–720
2. Pentland, A., Moghaddam, B., Starner, T.: View-based and modular eigenspaces for face recognition. In: Proc. of IEEE CVPR, Seattle, WA (1994)
3. Hastie, T., Tibshirani, R.: Discriminant analysis by Gaussian mixtures. Journal of the Royal Statistical Society series B **58** (1996) 158–176
4. Bach, F., Jordan, M.: A probabilistic interpretation of canonical correlation analysis. Technical Report 688, Department of Statistics, UC Berkeley (2005)
5. Fei-Fei, L., Fergus, R., Perona, P.: A bayesian approach to unsupervised one-shot learning of object categories. In: ICCV. (2003)
6. Tipping, M., Bishop, C.: Probabilistic principal component analysis. Technical Report NCRG/97/010, Neural Computing Research Group, Aston University. (1997)
7. Bishop, C.: Neural networks for pattern recognition. Oxford University Press (1995)
8. Sim, T., Baker, S., Bsat, M.: The cmu pose, illumination, and expression (pie) database. Proc. IEEE International Conference on Automatic Face and Gesture Recognition (2002)
9. Phillips, P., Wechsler, H., Huang, J., Rauss, P.: The feret database and evaluation procedure for face recognition algorithms. IVC **16**(5) (1998) 295–306
10. Nene, S., Nayar, S., Murase, H.: Columbia object image library: Coil. Technical Report CUCS-006-96, Department of CS, Columbia University (1996)
11. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290** (2000) 2323–2326
12. Tenenbaum, J.B., Freeman, W.T.: Separating style and content with bilinear models. Neural Computation **12**(6) (2000) 1247–1283
13. Mika, S., Ratsch, G., Weston, J., Scholkopf, B., Muller, K.: Fisher discriminant analysis with kernels. Proceedings of IEEE Neural Networks for Signal Processing Workshop (1999)