**LDA from "The elements of statistical learning" (Book, Hastie et al.)**

We want to learn the posterior probability where is the class and is the data. Suppose is the class-conditional density of . Let be the prior probability of class , with . Using Bayes theorem we get:

Naïve Bayes algorithm is a simple version of this analysis in which we assume that each class density is a product of marginal densities: the input data is conditionally independent in each class.

In LDA we assume that the density of each class follows a multivariate normal distribution:

We also assume that the covariance matrix of each class is the same. We then get:

Thus, we get that comparing two classes is a linear function of the data . The covariance matrix assumption cases the quadratic part and the constants to cancel. This results in a linear decision boundary between the two classes. The parameters are estimated from the data using MLE.

When we don't assume the covariance assumption we get a quadratic problem. The resulting classifier is usually similar to applying LDA in second-degree polynomial space (Figure 4.6, page 89). Regularized DA (RDA) proposes a tradeoff between LDA and QDA. Here we assume: .

Computing the LDA

We start by eigen-decomposition of the covariance matrices:, whereis a orthonormal matrix, and is a diagonal matrix with the positive eigenvalues. We can now transform the data to a sphere by using:

Here we used the common covariance assumption. And now classifications can be made based on distances from each class centroid, modulo the effect of the class priors. Further explanation: the linear transformation transforms the data to a new space in which the covariance matrix is diagonal; the second transformation equalize the variances in this new space; hence, the sphere.

Reduced-Rank LDA

The k centroids calculated in LDA lie in an affine subspace with a dimension size at most . Also, when we compute the distances, all distances orthogonal to this space contribute equally to all classes. Thus, we can use only this subspace in our classification and we do so without relinquishing any of the information needed for LDA. To find a subspace with lower dimension we can find the principal components of the centroids themselves.

The term affine indicates relation to geometry of affine space. The geometry is defined by a base that is not necessarily orthonormal. Given a vector space over a field , we define an affine space over the set . For every we get (subject to three simple conditions: uniqueness, sum with zero and order, see wolfram).

The first way to address this problem is based on Gaussian distributions:

1. Compute the class centroids (an matrix) and the common covariance matrix .
2. Compute the sphere data (using the decomposition of ).
3. Compute the between class covariance matrix. This is the covariance matrix of , denoted as . The decomposition defines the principal components, and can therefore be used for selecting an optimal subspace. The vectors define the coordinates of the new subspace.
4. Classify to the closest centroid in the reduced subspace. The value of the class is defined as:

The second way to address this problem was defined by Fisher. The goal is to find the linear combination such that the between-class variance (the variance between the class centroids) is maximized relative to the within class variance (the pooled variance of the centroids). So, let be the covariance matrix of the class centroids and let be the common covariance matrix we want to solve:

Question 1, equivalency between the two approaches: show that the optimal is identical to .

Let's start by showing that the covariance of a projected data is , where is the covariance of the original data and is the linear transformation.

The covariance of the transformed data is:

.

Question 2, classification rule: show that the LDA classifications rule is the same as performing the Gaussian classification with the constraint that centroids of the Gaussians lay in L-dimensional subspace.

**PCA vs. LDA (CS 479/679 Pattern Recognition – Spring 2006 course slides)**

Each dimensionality reduction technique finds a transformation of the data that maximizes some objective function. In PCA the goal is to reduce dimensionality while retaining as much as possible of the variation of the data.

PCA

The original space representation is and the new representation is , where . For example, we may find a new basis in which the coefficients of some coordinates are zero for all data points. In these cases these vectors in the basis can be removed and the dimension is lower.

The new space of PCA is centered at the sample mean and has directions determined by the "best" eigenvectors of the covariance matrix of the data . Because the covariance matrix is real and symmetric, the eigenvectors form an orthogonal basis.

The steps:

1. Center the data: .
2. The data matrix is now , calculate the covariance matrix
3. Compute the eigenvalues () and eigenvectors of (.
4. Since C is symmetric, the eigenvectors form a basis. The dimension reduction is performed by keeping the top K eigenvectors in the representation of the vectors in the new basis. In the new basis: . After dimensionality reduction we get: .

Geometric interpretation: PCA projects the data along the axis with the maximal variance (the eigenvalues). A simple threshold for selecting is the proportion of the variance explained by the new space. Formally we get: .

PCA is strongly dependent on the units used to measure the features (and their range). Thus, a common preprocessing step before applying PCA is to standardize each feature.

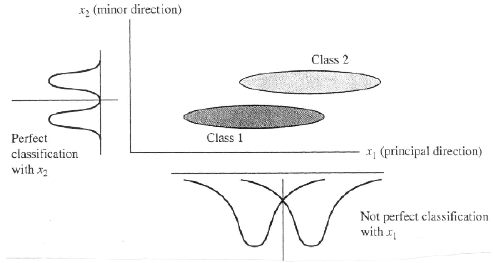
A case study from face detection (Turk and Pentland 1991):

1. Transform each image to a vector of size .
2. Subtract the mean face vector from each , yielding a new face vector . Let . A dimensions are .
3. Here we cannot calculate the covariance matrix explicitly as an is too much.
4. Instead of using for the covariance we use . What is the relation between the eigenvectors of , , and the eigenvectors of , ?
   1. .
   2. have the same eigenvalues .
   3. The eigenvalues of are the largest eigenvalues of .
   4. Use out of these eigenvectors for the dimensionality reduction.
5. The faces after stages 1-4 are called the eigenfaces.

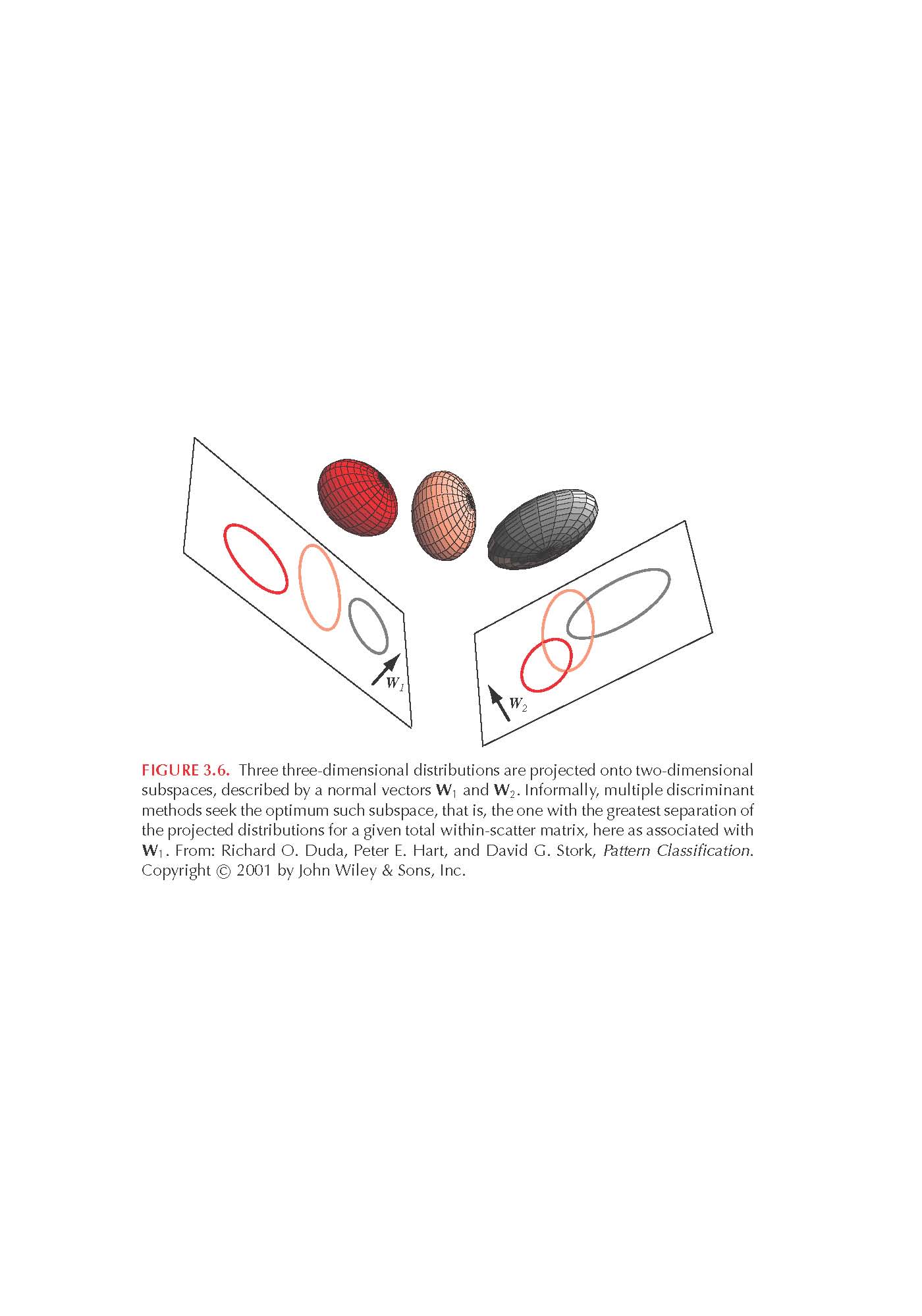
The distances in the new space are used to decide if a new image is a face. Technically, if the new image is close within a predefined threshold to one of the images in the trainings set than we classify the new image as a face.

PCA and classification

PCA is based on the covariance matrix that represents all samples and it does not account for class membership. The new space might reduce discriminative power (see figure below).



In LDA we perform dimensionality reduction while preserving as much of the discriminative power as possible. To do that we shall use the within-classes scatter and the between-classes scatter.



LDA

We are given classes. Let be the mean vector of class . Let be the number of samples of class and let be the total number of samples. Let be the mean vector of all samples.

1. Within-classes scatter matrix: .
2. Between-classes scatter matrix: . Since outer product of two vectors has a rank of at most 1, has a rank of at most . That is, the class means lay in an affine subspace of a dimension at most .

Let be the dimensionality reduction projection matrix (i.e., ). The goal in LDA is. Where the notation indicates scatter matrix of the projected data.

1. The solution for this problem is given by the vectors such that: . This problem is called a generalized eigenvector problem. The number of eigenvectors is at most . For representing the solution:

Hence, the optimum for the problem is:

If is non singular, the problem can be transformed to a standard eigenvalues problem. To achieve this property, we sometimes use PCA and then LDA in the PCA space.

A procedure for solving the generalized eigenvectors problem (from the SDA paper using their notations, and references 9, 17 within): we start by calculating the eigenvectors of and separately. We start by computing the eigenvectors of . We then look for that solves: :

Few words on determinant intuition:

1. The determinant is linear in each row. That is multiplying a row by a constant multiplies the determinant by the same constant.
2. Switching rows\columns changes the sign.
3. Consider the unit cube in N dimensional space: the set of vectors of length N with coordinates 0 or 1 in each spot. The determinant of the linear transformation (matrix) T is the signed volume of the region gotten by applying T to the unit cube. Example:
   1. Applying the unit transformation doesn't change the volume which is one.
   2. If you stretch the cube by a constant factor in one direction only, the new volume is that constant.
   3. And if you stack two blocks together aligned on the same direction, their combined volume is the sum of their volumes: this all shows that the signed volume we have is linear in each coordinate when considered as a function of the input vectors.
4. In multivariate calculus we use determinants (Jacobian) when we change coordinates in integration.

A direct LDA algorithm for High-Dimensional Data – with applications to face recognition

Our goal is to find a transformation that simultaneously diagonalizes both and :

To reduce dimensionality, we pick the top rows of (correspond to the top eigenvalues in ). The idea in this paper is that unlike the standard method we shall discard the null space by first diagonalizing and then . When is not singular, the methods are the same. But in high dimensional data this is not the usual case and the order changes the results. The steps of the algorithm:

1. Diagonalize : find such that . Where . This is a standard eigenvectors analysis. When we analyze data we immediately discard zero eigenvalues and their vectors (the Kernel). Dimensionality reduction is done by selecting the top rows of , (we get an matrix). We get: . is called the principal submatrix of .
2. Let . The transformation unites and reduces the dimension from to . We can now diagonalize :

.

Our objective is to maximize the ratio of the total scatter against the within-class scatter. Thus, we can look at the elements of and remove some of the **high** eigenvalues and their eigenvectors. These vectors correspond to high variability that is not discriminative.

1. Let the LDA matrix be: . It turns out that diagonalizes both components of the Fisher criterion: .
2. The final transformation that spheres the data is:

.

Discussion: The traditional analysis begins with diagonalizing . When is singular the eigenvectors with zero eigenvalues might be the most important features. Chen et al. (ref 2) showed that carries most of the discriminative power: for a projection direction , if , and then is maximized. The solution proposed in (ref 2) deals with this problem in the null space . The solution here addresses these problems as well and provides additional computational improvements (section 2.1).

**Fisher LDA (Max Welling’s summary note)**

The fisher criterion intuition: we want to separate the class means relative to the variance of the classes.

Our objective is: where: and is the covariance estimates within the classes: .

Pay attention that J is invariance to scale shift. Thus, we can constraint that the denominator is 1. The new problem is now:

The Lagrangian is: using KKT we get that the solution should hold: . This is a generalized eigen-decomposition problem.

When is positive definite our problem is equivalent to solving:

In the rest of the document: how to kernelize.

**Where are linear feature extraction methods applicable? Martinez and Zhu 2005**

Linear methods are popular in machine learning because:

1. When the linearity assumption holds, even few samples are enough for classification.
2. Most problems can be formulated in terms of generalized eigen-decomposition problems, for which many solvers are available.

Generalized eigen-decomposition problem (V and are the variables), **Equation (1)**:

We now get:

Hence, we get equation **(1)**:

Here is the matrix to maximize (between class scatter) and is the matrix to minimize (within class scatter). We know from the previous paper that the KKT conditions constraint that the solution for the Fisher-LDA will satisfy the equation of the generalized eigen-decomposition problem.

Hence, by the last formula we see that if we want to maximize the Fisher criterion then we need to select the top eigenvalues. Moreover we note that and are nonnegative matrices (covariance matrices) and therefore all eigenvalues above will not be negative.

One more important feature is that the rank of is at most the rank of (). , where is the matrix that holds the classes means, and therefore its maximal rank is minimum of the number of features and the number of classes . Since we usually center the data before running LDA or PCA the means of the classes are dependent and we lose one dimension. Thus, we can have at most nonnegative eigenvalues and that is the maximal dimension of the Fisher-LDA space.

Remark: under LDA assumptions we usually use the data covariance matrix instead of summing over the classes within covariance matrices.

In figure 2b we see an example in which the first eigenvector if PCA is the same eigenvector of but the eigenvalue of PCA is much larger. LDA selects the second axis even though the first one is better for classification. This simple example leads to **Theorem 1**: if then LDA does not necessarily minimize the Bayes error of the data. K measures the angle of each eigenvector of with the first eigenvectors of . These cases are called conflicts between the covariance to be maximized () and the covariance to be minimized ().

To prove the conflict statement more formally, we shall show that the discriminative power of (1) is equal to (TODO: CHANGE TO FORMULA)



First note that if is reversible than equation (1) is equivalent to:

This is a standard (non-generalized) eigen-decomposition problem. The discriminative power of the problem is the trace of which is the sum of the eigenvalues. Note that since the eigenvalues are nonnegative, maximizing the trace is equivalent to maximizing the determinant (the product of eigenvalues).