University of Cambridge

MPhil in Computer Speech Text & Internet Technology

Module: Speech Processing 1

Lecture 7: Linear Projection Schemes

Michaelmas 2002
Introduction

One of the classic problems encountered in pattern recognition is the curse of dimensionality. We have already (briefly) mentioned generalisation issues. In addition, some schemes that are practical in a low-dimensional space are impractical in high dimensional space.

Various techniques have been developed to reduce the dimensionality of the data. These projection schemes come in two classes

- **Supervised**: the class labels that we are trying to discriminate are know, e.g. *Fisher’s linear discriminant*.

- **Unsupervised**: the class labels are unknown, we only have the observations, e.g. *principal component analysis*

In this lecture we aim to map our original $d$-dimensional space down into a single dimension. Then a simple classifier may be generated in that single dimension (far simpler than dealing with our multiple dimensional classifiers). We will look at the underlying assumptions of the two schemes and how to implement them.
Fisher’s Linear Discriminant

Consider the simple 2-class problem shown above. It is clear that some projections yield better discrimination than others.

We intuitively feel that a good dimension is one in which the within-class variance is small and the between class variance is large. This is the basic aim of Fisher’s linear discriminant.

We need to express this intuition in a mathematical framework and then solve it!
Linear Projections

How do we mathematically represent a projection. The linear projection of a point $\mathbf{x}$ onto a line $\mathbf{w}$ may be expressed as

$$\tilde{x} = \mathbf{w}' \mathbf{x}$$

where $\tilde{x}$ represents the “projected” point. This comes from standard vector work.

The projection of the mean is similar. The mean in the projected space $\tilde{\mu}$ is given by

$$\tilde{\mu} = \frac{1}{n} \sum_{i=1}^{n} \tilde{x}_i = \frac{1}{n} \sum_{i=1}^{n} \mathbf{w}' \mathbf{x}_i$$

Since the projection is linear, we can write this as

$$\tilde{\mu} = \mathbf{w}' \left( \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \right) = \mathbf{w}' \mu$$

where $\mu$ is the vector mean of the original data. The projection of the covariance is more complex. It can be shown that

$$\tilde{\sigma}^2 = \mathbf{w}' \Sigma \mathbf{w}$$

where $\Sigma$ is the covariance matrix in the original space.

Check that you can prove this.
Fisher’s Discriminant Analysis

The aim is to choose the projection that maximises the distance between the class means, whilst minimising the within class variance. The cost function to *minimise* is

$$E(w) = -\frac{(\tilde{\mu}_1 - \tilde{\mu}_2)^2}{\tilde{s}_1 + \tilde{s}_2}$$

where $\tilde{s}_j$ is the projected scatter matrix for class $\omega_j$. The projected scatter matrix is defined as

$$\tilde{s}_j = \sum_{\tilde{x}_i \in \omega_j} (\tilde{x}_i - \tilde{\mu}_j)^2$$

From the previous slide this may be expressed as

$$E(w) = -\frac{w'S_Bw}{w'S_Ww}$$

where

$$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)'$$

and

$$S_W = S_1 + S_2$$

where

$$S_j = \sum_{x_i \in \omega_j} (x_i - \mu_j)(x_i - \mu_j)'$$

the mean of the class $\mu_j$ is defined as usual.
Finding 'Best” Projection

It can be shown that the vector $\hat{w}$ that minimises the cost function satisfies

$$S_B \hat{w} = \lambda S_W \hat{w}$$

This can be written as a standard eigenvalue problem

$$S_W^{-1} S_B \hat{w} = \lambda \hat{w}$$

Examining the term $S_B \hat{w}$, this can be written as

$$S_B \hat{w} = (\mu_1 - \mu_2)(\mu_1 - \mu_2)' \hat{w} = k(\mu_1 - \mu_2)$$

Therefore we know that the direction of the solution

$$\hat{w} = S_W^{-1}(\mu_1 - \mu_2)$$

This is known as Fisher’s linear discriminant direction.

This has allowed us to train the direction. However we still need to position the decision boundary in the projected space (effectively derive the $\tilde{w}_0$ term from the previous lecture. There are a variety of schemes that may be used:

- midway between the two means of the classes,
  $$\tilde{w}_0 = -\hat{w}'(\mu_1 + \mu_2)/2$$

- midway between the two points that are closest to the decision boundary (assuming separable data).
Example

Using the data from the previous lecture

\[ \mathbf{\mu}_1 = \begin{bmatrix} 0.825 \\ 0.500 \end{bmatrix}; \quad \mathbf{\mu}_2 = \begin{bmatrix} 0.1375 \\ 0.600 \end{bmatrix} \]

and

\[ \mathbf{S} = \begin{bmatrix} 0.2044 & 0.0300 \\ 0.0300 & 1.2400 \end{bmatrix} \]

So solving this yields

\[ \mathbf{w} = \begin{bmatrix} 3.3878 \\ -0.1626 \end{bmatrix} \]

This projection is shown above (with offsets in the y-axis to aid visualisation!).
Example (cont)

It is now necessary to generate a decision boundary. For this separable case the (negative value of the) midpoint between the boundary observations was used.

\[
\hat{a} = \begin{bmatrix} \hat{w} \\ \tilde{w}_0 \end{bmatrix} = \begin{bmatrix} 3.3878 \\ -0.1626 \\ -1.4432 \end{bmatrix}
\]

The decision boundary is shown above.

Confirm that you understand the direction of the decision boundary compared to \( \hat{w} \).
Principal Component Analysis

In contrast to Fisher’s discriminant analysis, principal component analysis (PCA) does not need the class labels. It is an unsupervised scheme. PCA uses the argument that with no additional information the most informative (useful) direction is the one with the greatest spread.

The first stage in this process is to remove the correlations in the training data. If we decorrelate the training data the covariance matrix may then be described by a diagonal covariance matrix and the one with the greatest spread is obvious. Decorrelating the data may be achieved using a linear transformation of the elements of the feature vector.

$$\tilde{x} = A'x$$

The covariance matrix in the transformed space is

$$\tilde{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} (\tilde{x}_i - \tilde{\mu})(\tilde{x}_i - \tilde{\mu})'$$

where we want $\tilde{\Sigma}$ to be a diagonal matrix. Rewriting the above equation

$$\tilde{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} A'(x_i - \mu)(x_i - \mu)'A = A'\Sigma A$$

where

$$\Sigma = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)(x_i - \mu)'$$

is the original covariance matrix.

We need the transform $A$ that achieves this decorrelation.
Eigenvectors/Eigenvalues

Eigenvectors and eigenvalues satisfy the following expression

\[ \mathbf{A} \mathbf{u}_i = \lambda_i \mathbf{u}_i \]

where \( \mathbf{u}_i \) is an eigenvector and \( \lambda_i \) is an eigenvalue of \( \mathbf{A} \). We will also require that \( \mathbf{u}_i' \mathbf{u}_i = 1 \).

The eigenvectors/eigenvalues may be found by finding all the values of \( \lambda \), the eigenvalues, that satisfy

\[ |\mathbf{A} - \lambda \mathbf{I}| = 0 \]

Having found the set of eigenvalues, we can find the set of eigenvalues by substituting in the original definition of an eigenvector/eigenvalue.

We can express a covariance matrix, \( \Sigma \), in terms of its eigenvectors and eigenvalues (note that \( \Sigma \) is real and symmetric)

\[ \Sigma = \mathbf{U} \Lambda \mathbf{U}' \]

where the matrix of eigenvectors is (note \( \mathbf{U}' = \mathbf{U}^{-1} \))

\[ \mathbf{U} = \begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_d \end{bmatrix} \]

and the matrix of eigenvalues is

\[ \Lambda = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_d \end{bmatrix} \]
Removing Correlations (2)

We can now substitute the eigenvector/eigenvalue expansion of the covariance matrix into the previous expression to get

\[ \tilde{\Sigma} = A'U\Lambda U'A \]

For a real symmetric matrix \( U' = U^{-1} \) so if we set \( A = U \) then

\[ \tilde{\Sigma} = U'U\Lambda U'U = \Lambda \]

But \( \Lambda \) is by definition a diagonal covariance matrix, so we have achieved the aim of decorrelating the data.

The covariance matrix after the transformation \( \tilde{\Sigma} \) is therefore

\[ \tilde{\Sigma} = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_d \end{bmatrix} \]

An alternative transformation is

\[ A = U\Lambda^{-\frac{1}{2}} \]

Substitute this in we get

\[ \tilde{\Sigma} = \Lambda^{-\frac{1}{2}}U'U\Lambda U'U\Lambda^{-\frac{1}{2}} = I \]
Example

Consider a two dimensional example

![Diagram of an ellipse being rotated by 30 degrees to a new axis.](image)

For the above distribution (ellipse is a line of constant probability density)

\[ \Sigma = \begin{bmatrix} 7 & \sqrt{3} \\ \sqrt{3} & 5 \end{bmatrix} \]

To find the eigenvalues solve

\[
\begin{vmatrix} 7 - \lambda & \sqrt{3} \\ \sqrt{3} & 5 - \lambda \end{vmatrix} = (7 - \lambda)(5 - \lambda) - 3 = (\lambda - 8)(\lambda - 4) = 0
\]

The eigenvalues are 8 and 4. To find the eigenvectors

\[
\begin{bmatrix} 7 & \sqrt{3} \\ \sqrt{3} & 5 \end{bmatrix} \mathbf{u}_1 = 8 \mathbf{u}_1
\]

solving this give

\[
\mathbf{u}_1 = \begin{bmatrix} \sqrt{3} \\ 2 \\ 1 \\ 2 \end{bmatrix}
\]

Similarly for \( \mathbf{u}_2 \) using \( \lambda = 4 \).
Example (cont)

We can now write the covariance matrix as

\[
\Sigma = U\Lambda U' = \begin{bmatrix}
\frac{\sqrt{3}}{2} & \frac{1}{2} \\
\frac{1}{2} & \frac{\sqrt{3}}{2}
\end{bmatrix} \begin{bmatrix} 8 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix}
\frac{\sqrt{3}}{2} & \frac{1}{2} \\
\frac{1}{2} & \frac{\sqrt{3}}{2}
\end{bmatrix}
\]

We can now find the decorrelating transform \( A \)

\[
A = U = \begin{bmatrix}
\frac{\sqrt{3}}{2} & \frac{1}{2} \\
\frac{1}{2} & \frac{\sqrt{3}}{2}
\end{bmatrix}
\]

The covariance matrix in the transformed space \( \tilde{\Sigma} \) is

\[
\tilde{\Sigma} = \Lambda = \begin{bmatrix} 8 & 0 \\ 0 & 4 \end{bmatrix}
\]

If we used the alternative transform

\[
A = U\Lambda^{-\frac{1}{2}} = \begin{bmatrix}
\frac{\sqrt{3}}{2} & \frac{1}{2} \\
\frac{1}{2} & \frac{\sqrt{3}}{2}
\end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{8}} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}
\]

This will result in \( \tilde{\Sigma} = I \).

Try \[
\begin{bmatrix}
3 & \sqrt{10} \\
\sqrt{10} & 6
\end{bmatrix}
\]
Principal Component Analysis

PCA is a method of extracting the most “important” features. It relies on the assumption that the direction in which there is most variation in the data is the most discriminatory (contains most class information). Since the variation in the data is described by the covariance matrix, we want the directions in which the variances are the largest.

The simplest way to obtain these is to decorrelate the data and use the eigenvectors with the largest eigenvalues. As usual the covariance matrix of the data may be written as

$$\Sigma = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu)(x_i - \mu)'$$

where $\mu$ is the global mean of the data. We can decorrelate $\Sigma$ using its eigenvalues and eigenvectors,

$$\Sigma = U\Lambda U'$$

We know that the covariance matrix in the transformed space using $A = U$ is a diagonal covariance matrix whose elements on the leading diagonal are the eigenvalues (ordered in this case in decreasing value). If we select the top $p$ eigenvalues we have the $p$-dimensions with the greatest variability. Thus the $d \times p$ projection transformation matrix, $A^{(p)}$ is

$$A^{(p)} = U \begin{bmatrix} I^{(p)} \\ 0 \end{bmatrix}$$
PCA (cont)

The $p$-dimensional feature vector is

$$\tilde{x}^{(p)} = A^{(p)'} x$$

The main problem with PCA is that the eigenvalues/vectors are not independent of scaling. If one dimension is scaled then this affects the eigenvalues and the eigenvectors selected. However simply scaling a dimension does not increase the power of that dimension to discriminate.
Simple Example (again)

Again using the data from the previous lecture we can find the direction of maximum variance. From the data we find

\[
\Sigma = \begin{bmatrix} 0.144 & -0.013 \\ -0.013 & 0.158 \end{bmatrix}
\]

\[
= \begin{bmatrix} 0.521 & 0.853 \\ -0.853 & 0.521 \end{bmatrix} \begin{bmatrix} 0.166 & 0 \\ 0 & 0.136 \end{bmatrix} \begin{bmatrix} 0.521 & -0.853 \\ 0.853 & 0.521 \end{bmatrix}
\]

Using the principal direction and positioning the boundary mid-way between the means gives the following decision boundary.

Not optimal!!