Speaker Recognition in non-linear signal processing and pattern recognition. Change title. Group 6

Line Aggerbo, Rasmus S. Reimer & Jonas D. Pedersen

Aarhus University, Department of Engineering

Abstract

The Content of this paper seeks to present the knowledge gained throughout the non-linear signal processing and pattern recognition course from Aarhus University, department of engineering. The paper is split into multiple sections explaining the data used in the paper, the methods used to process the data and the methods used for categorising the data. The results are presented along with a discussion of the results. Rewrite the abstract

I. Dimensionality Reduction

Remove this section from the report.
When working with large amounts of data in higher dimensions you often need a lot of computing power. Dimensionality reduction is mostly used in order to speed up the learning algorithm. This section contains an explanation of two widely used dimensionality reduction methods, Principal component analysis (PCA) and Fisher Discriminant method (Fisher).

I. PCA

PCA is used for reducing the number of dimensions of a feature space. It works by projecting the data in the feature space, down to a fewer dimensional feature space by minimizing the squared projection error. The reduced feature space does not nessecarily share the same features, but new features are found which best retains the variance in the data.
PCA should mainly be used for compressing the data to save memory or reducing running time of learning algorithm. By reducing the amount of features most machine learning algorithms runs faster. PCA can also be used to prevent overfitting, but it is usually better to use regularization.
Figure 1 shows a 3 dimensional feature space where all the data, within a small margin, lies in a 2 dimensional plane. PCA is used to find two vectors $u^{(1)}$ and $u^{(2)}$ which spans this 2D plane. The plane is the 2 dimensions, in which the most variance is obtained in the data. Preprocessing of the data should be done before doing PCA. Given the training set:

$$x = \begin{bmatrix} x_1 & x_2 & \dots & x_m \end{bmatrix} \tag{1}$$

Ensure that every feature has zero mean by doing mean normalization:

$$\mu_j = \frac{1}{m} \sum_{i=1}^{m} x_j^{(i)} \tag{2}$$

$$x_j = x_j - \mu_j \tag{3}$$

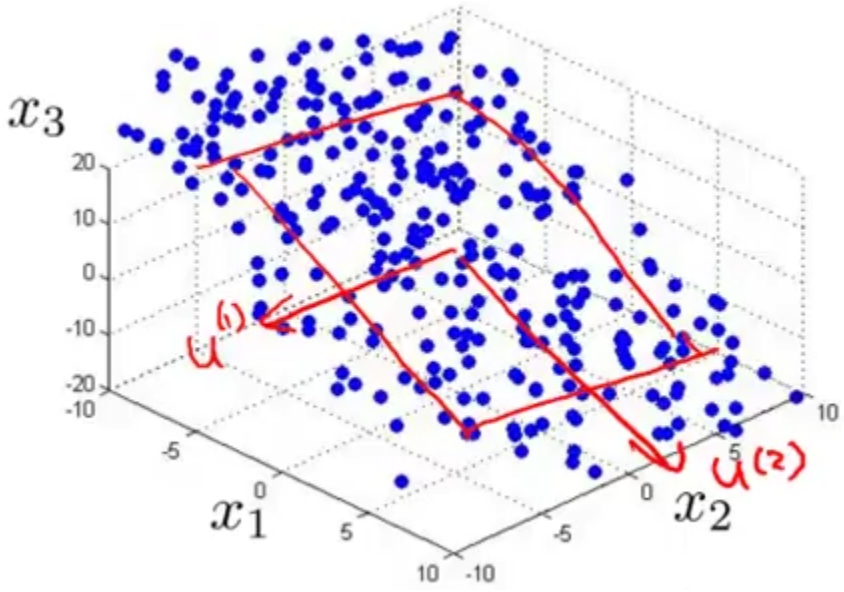

Figure 1: *3D to 2D PCA illustration*

After preprocessing the data, we can do PCA on it. We start by computing the covariance matrix Σ:

$$\Sigma = \frac{1}{m} \sum_{i=1}^{n} x^{(i)} {x^{(i)}}^T \tag{4}$$

The covariance matrix describes how the different features relates. When doing feature reduction we want to remove features which has high correlation with other features. An example could be a feature which describes a length in cm and another feature describing the same length in inches. These features will have very high correlation and one of them can be removed from the feature space without loosing much information.
Then we compute the eigenvectors of covariance matrix:

$$U = \begin{bmatrix} u^{(1)} & u^{(2)} & \dots & u^{(n)} \end{bmatrix} \in \mathbb{R}^{n \times n} \tag{5}$$

The eigenvectors will lay in the directions of most variance in the data. This is what is shown on Figure 1. The longer the eigenvector, the more variance it describes. Therefore we want to keep the longest eigenvectors and remove the shortest eigenvectors. The eigenvectors are ordered by length in the matrix U. We select the first k eigenvectors to get the reduced set of eigenvectors:

$$U_{reduce} = \begin{bmatrix} u^{(1)} & u^{(2)} & \dots & u^{(k)} \end{bmatrix} \tag{6}$$

We can now calculate the new feature vectors:

$$z = U_{reduce}^T x \tag{7}$$

We have now reduced the feature space to a k dimensional feature space. Say we want to retain at

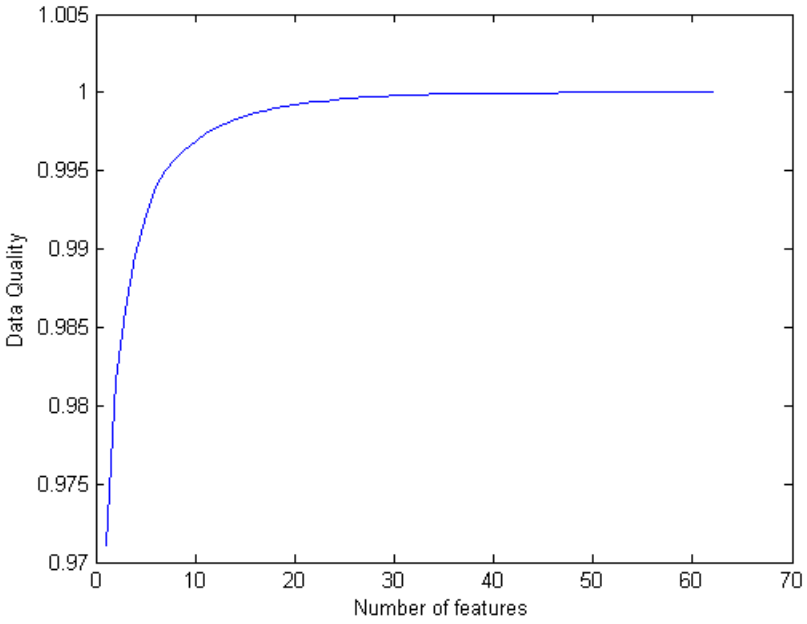

Figure 2: *Retained variance per dimension*

least 95% of the variance in the data. We do this by picking the smallest value of k so that:

$$\frac{\sum_{i=1}^{k} S_{ii}}{\sum_{i=1}^{n} S_{ii}} \geq 0.95 \tag{8}$$

The matrix S is found by doing singular value decomposition (SVD). The matrix S has the form:

$$S = \begin{bmatrix} S_{11} & 0 & 0 & 0 & 0 \\ 0 & S_{22} & 0 & 0 & 0 \\ 0 & 0 & S_{33} & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & S_{nn} \end{bmatrix} \tag{9}$$

In our project we use PCA to reduce our feature space from 64 dimensions down to 40 dimensions. We do this to increase the speed of our learning algorithms while still retaining almost all of our data ($\geq$ 99.99%) as seen on Figure 2.

II. Fisher

Fisher's linear discriminant is another look on dimensionality reduction. Instead of maximizing the variance of the data, we use the information we have on the classes to separate them as much as possible. We use equation 10 and place a threshold on y such that class 1: $y \geqslant -\omega_0$ and class 2 is everything else. This is also said as projecting data down to one dimension.

$$y = \mathbf{w}^T \mathbf{x} \tag{10}$$

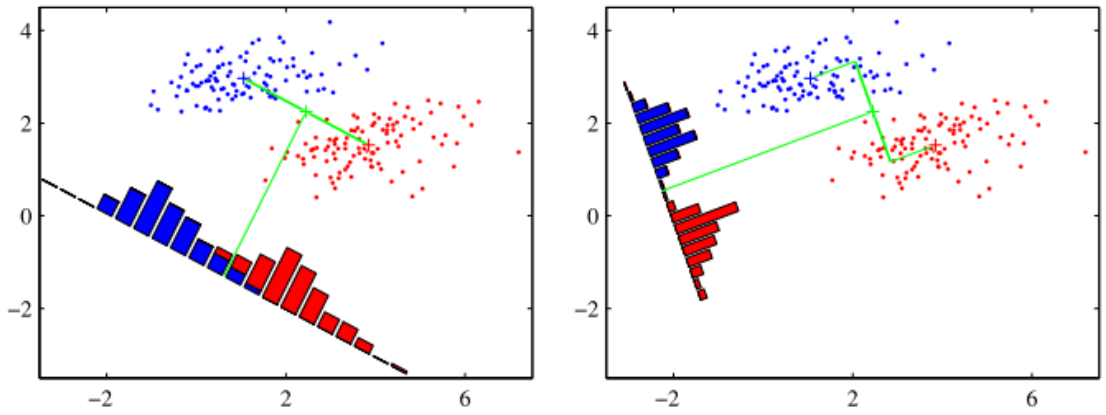

Figure 3: *Picture from the Pattern Recognition and Machine Learning Book by Christopher M. Bishop*

This leads to a considerable loss in information and may cause overlapping in data that did not overlap in multidimensional space as can be see on figure 3 from the book Pattern Recognition and Machine Learning by Christopher M. Bishop[1]. The left Image is the original space while the right image is in the projected space.
By adjusting the weight vector **w** a solution can be found that minimises overlapping by maximising the distance between classes. Given two classes we have:

$$m_1 = \frac{1}{N_1} \sum_{n=C_1} X_n, m_2 = \frac{1}{N_2} \sum_{n=C_2} X_n \tag{11}$$

With **m** being the mean of the class and `N` being the amount of points in the class. The goal is to maximise the difference $m_2 - m_1$ and this is done by maximising the difference of the projected data as well:

$$m_2 - m_1 = \mathbf{w}^T \times (\mathbf{m}_2 - \mathbf{m}_1) \tag{12}$$

We add the constraint that $\sum_i \omega_i^2 = 1$ in order to avoid large expressions when dealing with the projected data. When solving this we see that there might be an issue with considerable overlap in the projected space.
The fisher method seeks to minimise the class overlap by reducing in class variance. Within-class variance is given by:

$$s_k^2 = \sum_{n=C_1} (y_n - m_k)^2 \tag{13}$$

where $y_n = \mathbf{w}^T \mathbf{x}_n$ and m_k is the difference mentioned earlier. The fisher criterion is given by:

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \tag{14}$$

This can also be written as:

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} \tag{15}$$

Where $\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T$ is the between-class covariance matrix and
$\mathbf{S}_W = \sum_{n \in C_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^T + \sum_{n \in C_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^T$ is the within-class covariance

matrix. Differentiating and removing the scaling factors we get:

$$\mathbf{w} \propto \mathbf{S}^{-}1_W(\mathbf{m}_2 - \mathbf{m}_1) \tag{16}$$

This is know as Fisher's linear discriminant. This holds for 2 classes but in this project there is 3 classes. A general term for Fisher's discriminant for more than two classes must be considered. Given $\mathbf{y} = \mathbf{W}^T\mathbf{x}$ where $\mathbf{W}$ is the weight vectors $\mathbf{w}_k$. The within-class covariance for K classes is given by:

$$\mathbf{S}_W = \sum_{n=1}^{K} \sum_{n=C_k} (\mathbf{x}_n - \mathbf{m}_k)(\mathbf{x}_n - \mathbf{m}_k)^T \tag{17}$$

With $\mathbf{m}_k = \frac{1}{N_k} \sum_{n=C_k} \mathbf{x}_n$, N_k is the number of patterns in class k. The total covariance matrix given by Duda and Hart(1973) is used to find the between-class covariance matrix.

$$\mathbf{S}_T = \sum_{n=C_k}^{N} (\mathbf{x}_n - \mathbf{m})(\mathbf{x}_n - \mathbf{m})^T \tag{18}$$

where $\mathbf{m}$ is the mean of the total data set. The between class covariance matrix can be found by using $\mathbf{S}_T = \mathbf{S}_W + \mathbf{S}_B$.

$$\mathbf{S}_B = \sum_{n=C_k}^{N} N_k(\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^T \tag{19}$$

Defining these matrices in the projected space we get:

$$\mathbf{s}_W = \sum_{n=1}^{K} \sum_{n=C_k} (\mathbf{y}_n - \boldsymbol{\mu}_k)(\mathbf{y}_n - \boldsymbol{\mu}_k)^T \tag{20}$$

$$\mathbf{s}_B = \sum_{n=C_k}^{N} N_k(\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^T \tag{21}$$

where $\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=C_k} \mathbf{y}_n$ and $\boldsymbol{\mu} = \frac{1}{N_k} \sum_{k=1}^{K} N_k\boldsymbol{\mu}_k$.
The cost function is then defined as:

$$J(\mathbf{W}) = Tr\{\mathbf{s}_W^{-1}\mathbf{s}_B\} \tag{22}$$

The means and covariances are estimated from the training set. The resulting dimensions is K - 1 where K is the number of classes. In this project the dimensions of the fisher linear discriminant is $3 - 1 = 2$. Analysing these projected features does not provide distinct classes in this project. An examples of this is found in the linear classifier section when plotting 2 dimensional features.

References

[1] Christopher Bishop. *Pattern recognition and machine learning*. Springer, New York, 2006.

Remove "Pattern recognition and machine learning" from references.