CSMMA16

Mathematics and Statistics

PCA and LDA

30th September 2016

PCA

Examples in R

LDA

QDA

Introduction to PCA

Principal component analysis (PCA) is a variable reduction technique that uses a linear transformation to construct a smaller number of artificial variables (called principal components) from a larger number of variables,

PCA has found applications in areas such as computer vision (face recognition), data mining (big data analytics)and image processing and compression.

Consider the matrix

$$\mathbf{X} = \begin{pmatrix} \mathbf{x_1} & \mathbf{x_2} & \cdots & \mathbf{x_k} \end{pmatrix}$$

where $\mathbf{x_j}$ is $n \times 1$ column vector with i^{th} element $x_{i,j}$ the i^{th} *centered* observation of the variable $\mathbf{X_j}$, $j = 1, \ldots, k$.

Principal components directions

The sample variance-covariance matrix $\mathbf{S}^2 = \mathbf{X}^T\mathbf{X}/(n-1)$ and the eigen decomposition of $\mathbf{X}^T\mathbf{X}$ (and hence of $\mathbf{S}^2$, up to a factor of n) can be written

$$\mathbf{X}^T\mathbf{X} = \mathbf{U}\mathbf{D}^2\mathbf{U}^T$$

where D is the diagonal matrix with diagonal entries the ordered singular values $\sqrt{\lambda_1} \geq \sqrt{\lambda_2} \geq \ldots \geq \sqrt{\lambda_k}$ of $\mathbf{X}$ and

$$\mathbf{U} = \begin{pmatrix} \mathbf{u_1} & \mathbf{u_2} & \cdots & \mathbf{u_k} \end{pmatrix}$$

is matrix of normalised eigen-vectors of $\mathbf{X}^T\mathbf{X}$.

The $\mathbf{u_j}$, $j = 1, \ldots, k$ are also called the principal components directions of $\mathbf{X}$.

Principal components

Consider the transformation $\mathbf{Z}$ of $\mathbf{X}$ given by

$$\mathbf{Z} = \mathbf{X}\mathbf{U} = \begin{pmatrix} \mathbf{X}\mathbf{u}_1 & \mathbf{X}\mathbf{u}_2 & \cdots & \mathbf{X}\mathbf{u}_k \end{pmatrix}$$

Each column $\mathbf{z}_j$ of $\mathbf{Z}$ is the weighted linear combination of the original vectors of observations

$$\mathbf{z}_j = \mathbf{X}\mathbf{u}_j = \mathbf{x}_1 u_{1,j} + \mathbf{x}_2 u_{2,j} + \cdots + \mathbf{x}_k u_{k,j}$$

and as the eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k$ are orthogonal, $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k$ are uncorrelated.

Further the sample variance s_j^2 of $\mathbf{z}_j$ is

$$s_j^2 = \frac{\mathbf{u}_j^T \mathbf{X}^T \mathbf{X} \mathbf{u}_j}{n-1} = \frac{\mathbf{u}_j^T \lambda_j \mathbf{u}_j}{n-1} = \frac{\lambda_j}{n-1}$$

and as we set $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k$, it follows that $s_1^2 \geq s_2^2 \geq \ldots \geq s_k^2$

The derived variables $\mathbf{z}_j,\ j = 1, \ldots, k$ are called the principal components of $\mathbf{X}$.

Characteristics of principal components

Principal components (scores) are constructed by accounting for observed variance in the data

- The first component $\mathbf{z}_1$ accounts for the largest proportion of the total variance
  - may therefore be correlated with many of the observed variables
- The second component $\mathbf{z}_2$ accounts for a maximal amount of variance not accounted for by first component and
  - is uncorrelated with the first component: $\rho(\mathbf{z}_1, \mathbf{z}_2) = 0$
- The third component $\mathbf{z}_3$ accounts for a maximal amount of variance not accounted for by first and second components and
  - is uncorrelated with the first and second components: $\rho(\mathbf{z}_1, \mathbf{z}_3) = \rho(\mathbf{z}_2, \mathbf{z}_3) = 0$
- and so on ...

Characteristics of principal components

The number of components equals the number of observed variables but

- each new component accounts for progressively smaller and smaller amounts of variance of the variables and
- usually only the first few account for meaningful amounts of variance
- so often only the first few components are used

Note that some authors refer to the eigen vectors as the principal components

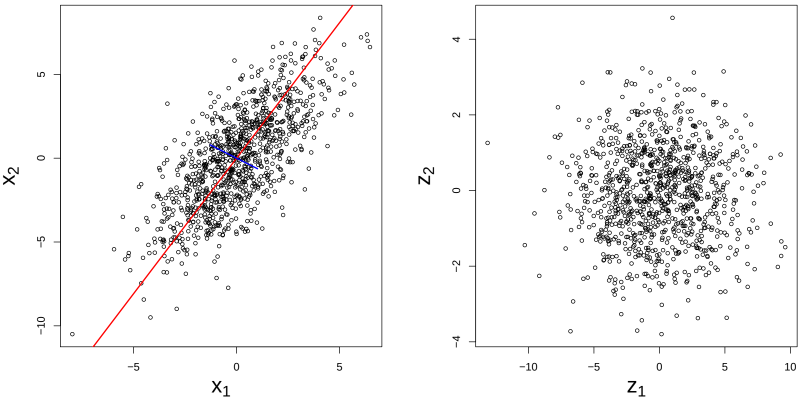
Illustration

```
x1<-rnorm(1000,0,2)
x2<-x1+rnorm(1000,0,2)
eig<-eigen(crossprod(cbind(x1,x2)))
U<-eig$vectors
lam<-eig$values
Z<-cbind(x1,x2)%*%U

> U
          [,1]       [,2]
[1,] 0.5259376 -0.8505232
[2,] 0.8505232  0.5259376

> lam
[1] 11287.879  1563.738
```

Plots of x_1, x_2 and z_1, z_2 are shown on the next slide.



Example

Results of the Olympic heptathlon competition, Seoul, 1988:
A Handbook of Statistical Analyses Using R (3rd Edition)
25 competitors in 7 events: hurdles, highjump, shotput, run200m, longjump, javelin, run800m

```
install.packages("HSAUR3")
library(HSAUR3)
head(heptathlon)
```

```
                    hurdles highjump  shot ...  run800m score
Joyner-Kersee (USA)    3.73     1.86 15.80 ...    34.92  7291
John (GDR)             3.57     1.80 16.23 ...    37.31  6897
...                    ...      ...   ...  ...     ...    ...
```

Ignoring last competitor:

```
X<-heptathlon[-25,1:2]
```

HSAUR3

Results for the seven events on different scales so compute principal components from correlation matrix.

Can centre and scale each variable and use the eigen function (as in the illustration)

Alternatively:

```
X.pca<-prcomp(X[,-which(colnames(X) == "score")],scale=T)
```

The prcomp() function returns a list including:

- sdev - standard deviations of the principal components (the singular values of $X^T X$ divided by $\sqrt{n-1}$)
- rotation - matrix of variable loadings (eigenvectors)
- center, scale - the centering and scaling used

E.g:

```
X.pca$sdev, X.pca$rotation, X.pca$scale
```

Data classification

Often necessary (useful) to classify (group) data into categories (sub-populations).

Example: Disease Diagnosis

Placing a patient presenting a set of symptoms (vector of data) into one of two classes, ill or not ill.

Linear data classification methods:

- uses linear (discriminant) functions to separate/classify data into groups/classes
- thus boundaries separating classes (decision boundaries) are linear

Introduction to LDA

Let $\mathbf{X} \in \mathbb{R}^p$ be a random vector and G a random categorical variable and suppose we wish to group observations (realisations) $\mathbf{x}_1, \ldots, \mathbf{x}_n$ of $\mathbf{X}$ into classes of G.

We can achieve optimal classification if we know the probabilities of $\mathbf{x}$ belonging to classes of G:

$$P(G|\mathbf{X} = \mathbf{x})$$

We simply classify to the most probable class.

Let $f_k(\mathbf{x})$ denote the conditional density of $\mathbf{X}$ given $G = k$ and let $\pi_k = P(G = k),\ k = 1, \ldots, K$.

By Bayes Theorem

$$P(G = k|\mathbf{X} = \mathbf{x}) = \frac{f_k(\mathbf{x})\pi_k}{\sum_{\ell=1}^{K} f_\ell(\mathbf{x})\pi_\ell}$$

This shows our ability to classify depends on how well we know $f_k(\mathbf{x})$

LDA

Suppose we assume each class density $f_k(\mathbf{x})$ is multivariate normal with mean vector $\boldsymbol{\mu}_k$ and variance-covariance matrix $\boldsymbol{\Sigma}_k$,

$$f_k(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^p|\boldsymbol{\Sigma}_k|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)}$$

Linear discriminant analysis (LDA) arises in the special case when we assume classes have same covariance matrix, i.e. $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma},\ \forall k$

In LDA we compare membership of two classes k and ℓ by the log-ratio

$$\log \frac{P(G = k|\mathbf{X} = \mathbf{x})}{P(G = \ell|\mathbf{X} = \mathbf{x})} = \log \frac{f_k(\mathbf{x})}{f_\ell(\mathbf{x})} + \log \frac{\pi_k}{\pi_\ell}$$

which simplifies to the linear equation in $\mathbf{x}$,

$$= \log \frac{\pi_k}{\pi_\ell} - \frac{1}{2}(\boldsymbol{\mu}_k + \boldsymbol{\mu}_\ell)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_\ell) + \mathbf{x}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_k - \boldsymbol{\mu}_\ell)$$

From this, we can see the linear discriminant functions are

$$\delta_k(\mathbf{x}) = \log(\pi_k) - \frac{1}{2}\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \mathbf{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k$$

In practice the parameters $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}$ are unknown and are estimated from training data.

- $\hat{\pi}_k = \frac{n_k}{n}$ where n_k is number of observations in k^{th} class
- $\hat{\boldsymbol{\mu}}_k = \sum_{g_i=k} \frac{\mathbf{x}_i}{n_k}$ where g_i genotes group of $\mathbf{x}_i$
- $\hat{\boldsymbol{\Sigma}} = \sum_{k=1}^{K} \sum_{g_i=k} \frac{(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)(\mathbf{x}_i - \hat{\boldsymbol{\mu}}_k)^T}{n-K}$

The decision rule is that a new observation $\mathbf{x}$ is placed in the class for which

$$\delta_k(\mathbf{x}) = \log(\hat{\pi}_k) - \frac{1}{2}\hat{\boldsymbol{\mu}}_k^T \hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{\mu}}_k + \mathbf{x}^T \hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{\mu}}_k$$

is a maximum

QDA

Assume that classes have different covariance matrices.

It follows from the multivariate normal density that the discriminant function

$$\delta_k(\mathbf{x}) = -\frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{x} - \boldsymbol{\mu}_k) + \log(\pi_k)$$

is now a quadratic function.

This is quadratic discriminant analysis (QDA).

- Notice that QDA (and LDA) finds the centroid of classes from training data and then classifies a new observation by finding the closest centroid to it
- but also takes account of the correlation structure when defining distance

Computations for LDA and QDA

Computations are simplified by diagonalising $\hat{\boldsymbol{\Sigma}}_k$ (for QDA) or $\hat{\boldsymbol{\Sigma}}$ (for LDA).

Writing $\hat{\boldsymbol{\Sigma}}_k = \mathbf{U}_k \mathbf{D}_k \mathbf{U}_k^T$, we get

$$\hat{\delta}_k(\mathbf{x}) = -\frac{1}{2}\sum_{\ell} \log(d_{kl}) - \frac{1}{2}\left[\mathbf{U}_k^T(\mathbf{x}-\boldsymbol{\mu}_k)\right]^T \mathbf{D}_k^{-1}\left[\mathbf{U}_k^T(\mathbf{x}-\boldsymbol{\mu}_k)\right] + \log(\pi_k)$$

For LDA transform the data $\mathbf{X}$ using the eigen decomposition of $\hat{\boldsymbol{\Sigma}} = \mathbf{U}\mathbf{D}\mathbf{U}^T$

$$\mathbf{X}^* = \mathbf{D}^{-\frac{1}{2}}\mathbf{U}^T\mathbf{X}$$

The common variance-covariance estimate of $\mathbf{X}^*$ is the identity matrix $\mathbf{I}$.

Classify new data to closest class centroid in the transformed space after adjusting for the probabilities π_k

Vowel sound recognition

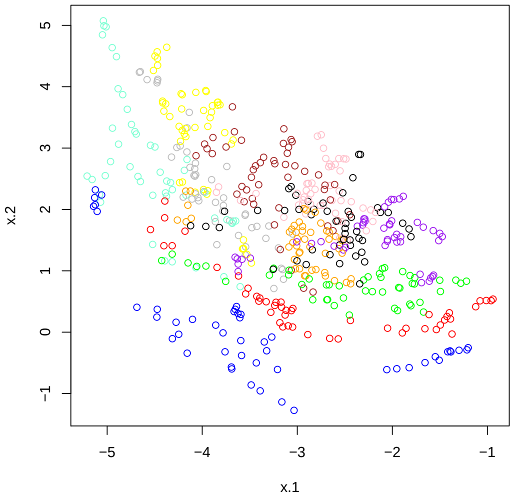
Speech signals from different speakers of 11 vowel sounds were digitised and for each signal 10 log area ratios (lar) were derived from linear prediction coefficients (lpc)

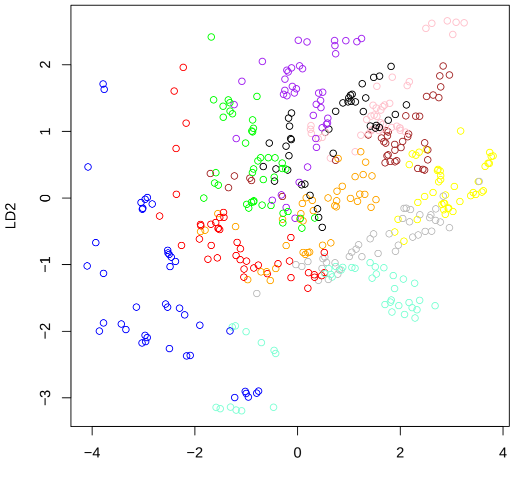
Want to test use of the 10 measures for classifying sounds from independent speakers

```
y    x.1   x.2    x.3   x.4    x.5   x.6    x.7    x.8    x.9   x.10
1 -3.639 0.418 -0.670 1.779 -0.168 1.627 -0.388  0.529 -0.874 -0.814
2 -3.327 0.496 -0.694 1.365 -0.265 1.933 -0.363  0.510 -0.621 -0.488
3 -2.120 0.894 -1.576 0.147 -0.707 1.559 -0.579  0.676 -0.809 -0.049
4 -2.287 1.809 -1.498 1.012 -1.053 1.060 -0.567  0.235 -0.091 -0.795
.....................................................................
```

```
library(MASS)
z <- lda(y~x.1+x.2,dat,prior = rep(1,11)/11)
y.pred<-predict(z,dat2[,2:3]) #dat2 is test data
```

```
##coefficients of the two (max(2,11-1)) discriminant functions
z$scaling
             LD1       LD2
x.1 0.09508135 1.6595441
x.2 1.49930291 0.6507017
```





PCA and LDA

- Both LDA and PCA are linear transformation methods
- PCA yields the directions that maximizes the variance of the data
- LDA finds the directions that maximizes the separation (or discrimination) between different classes

PCA projects the entire dataset onto a different space while LDA tries to construct a space that distinguishes between patterns of the different classes.