Lecture 2d

Latent variable models

1

Overview

- Principal Component Analysis (PCA)
- Probabilistic PCA
- Independent component analysis (ICA)

2

Latent variables

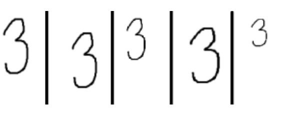
- Sometimes data depends on the variables we can not measure (hard to measure)
  - Answers on the test depend on Intelligence
  - Brain activity in the brain is measured by sensors
  - Stock prices depend on market confidence

X ← Z

3

Latent variables

- Latent factor discovered → data storage may decrease a lot



- Latent factors
  - Center
  - Scaling
- Original vs compressed
  - 100x100x5=50000
  - 100x100+2*5+2*5=10020

4

Principal Component Analysis (PCA)

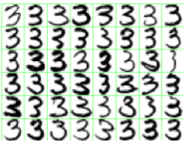
- *PCA* is a technique for reducing the complexity of high dimensional data
- It can be used to approximate high dimensional data with a few dimensions (latent features) –> much less data to store
- New variables might have a special interpretation

Applications

- Image recognition
- Information compression
- Subspace clustering
- …

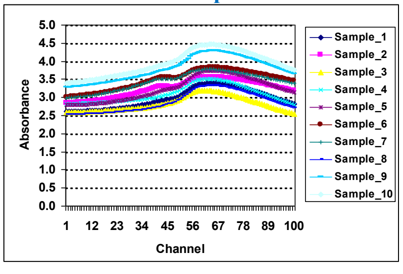
Principal Component Analysis (PCA)

- Example 1: Hadwritten digits
  - Can we get a more compact summary?



Absorbance records for ten samples of chopped meat

Parallel coordinate plot for "FAT"



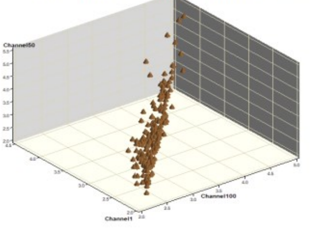
1 target (fat)

100 features (absorbance at 100 wavelengths or channels)

The features are strongly correlated to each other

3-D plots of absorbance records for samples of meat - channels 1, 50 and 100

Scatterplot for three components



Principal components analysis

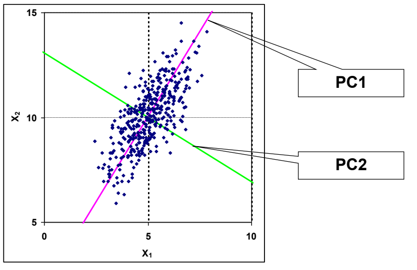
Idea: Introduce a new coordinate system (PC1, PC2, …) where

- The first principal component (PC1) is the direction that maximizes the variance of the projected data
- The second principal component (PC2) is the direction that maximizes the variance of the projected data after the variation along PC1 has been removed
- The third principal component (PC3) is the direction that maximizes the variance of the projected data after the variation along PC1 and PC2 has been removed
- ….

In the new coordinate system, coordinates corresponding to the last principal components are very small → can take away these columns

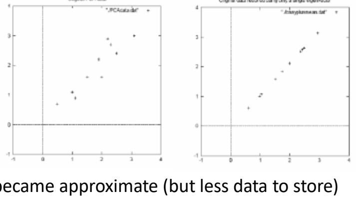
9

Principal Component Analysis - two inputs



10

PCA- after reducing dimensionality



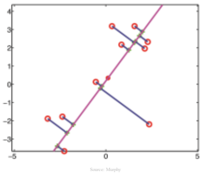
- Data became approximate (but less data to store)
- $PC_1, \dots PC_M$ are actually eigenvections of **sample covariance** (first largest eigenvalue,…,Mth largest egenvalue)

11

PCA: another view

- Aim: minimize the distance between the original and projected data

$$\min_V \sum_{i=1}^{N} \|x_n - \tilde{x}_n\|^2$$



12

PCA: computations

Data $D = \left\| \mathbf{x}_1\ \mathbf{x}_2 \ldots \mathbf{x}_p \right\|, \quad \mathbf{x}_i = (x_{i1}, \ldots, x_{in})$

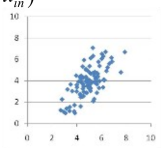
1. Centred data

$$X = \left\| \mathbf{x}_1 - \bar{\mathbf{x}}_1\ \mathbf{x}_2 - \bar{\mathbf{x}}_2 \ldots \mathbf{x}_p - \bar{\mathbf{x}}_p \right\|,$$

2. Covariance matrix

$$\boldsymbol{S} = \frac{1}{N} \boldsymbol{X}^T \boldsymbol{X}$$

3. Search for eigenvectors and eigenvalues of **S**

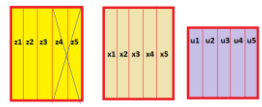


	Column 1	Column 2
Column 1	0.951	0.905
Column 2	0.905	1.883

13

PCA: computations

4. Coordinates of any data point $x=(x_1 \ldots x_p)$ in the new coordinate system:
$z = (z_1, \ldots z_n)$, $z_i = x^T u_i$



Matrix form: $Z = X\,U$

5. Discard principle components after some *M*

6. New data will have dimensions N x M instead of N x p

Getting approximate original data:

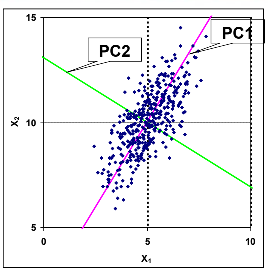
$$X' = Z U_M^T$$

Store: N x M+ p x M instead N x p

100*50 vs
100*4+50*4

14

Principal Component Analysis



Eigenanalysis of the Covariance Matrix

Eigenvalue	2.8162	0.3835
Proportion	0.880	0.120
Cumulative	0.880	1.000

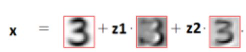
Variable	PC1	PC2
X1	0.523	0.852
X2	0.852	-0.523

Loadings (U)

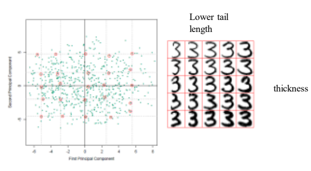
15

Principal Component Analysis

- Digits: two eigenvectors extracted



- Interptretation of eigenvectors



16

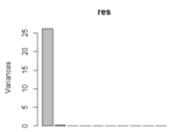
PCA in R

- Prcomp(), biplot(), screeplot()

```
mydata=read.csv2("tecator.csv")
data1=mydata
data1$Fat=c()
res=prcomp(data1)
lambda=res$sdev^2
#eigenvalues
lambda
#proportion of variation
sprintf("%2.3f",lambda/sum(lambda)*100)
screeplot(res)
```

```
> lambda
 [1] 2.612713e+01 2.385369e-01 7.844883e-02 3.018501e-
 [7] 2.052212e-04 1.084213e-04 2.077326e-05 1.150359e-
```

```
> sprintf("%2.3f",lambda/sum(lambda)*100)
 [1] "98.679" "0.901"  "0.296"  "0.114"  "0.006
 [9] "0.000"  "0.000"  "0.000"  "0.000"  "0.000
```



Only 1 component captures the 99% of variation!

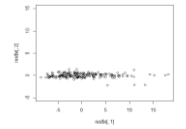
17

PCA in R

- Principal component **loadings (U)**

```
U=res$rotation
head(U)
```

```
> head(U)
                PC1       PC2        PC3
Channel1 0.07938192 0.1156228 0.08073156 -0.0927
Channel2 0.07987445 0.1170972 0.07887873 -0.0981
Channel3 0.08036498 0.1185571 0.07702127 -0.1031
Channel4 0.08085611 0.1200006 0.07515015 -0.1077
Channel5 0.08135022 0.1214075 0.07323819 -0.1119
Channel6 0.08184806 0.1227401 0.07125048 -0.1156
```



- Data in (PC1, PC2) – **scores (Z)**

plot(res$x[,1], res$x[,2], ylim=c(-5,15))

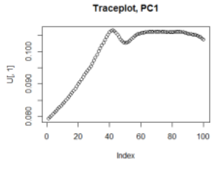
Do we need second dimension?

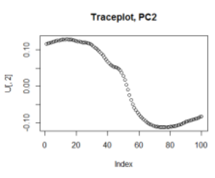
18

PCA in R

- Trace plots

```
U=loadings(res)
plot(U[,1], main="Traceplot, PC1")
plot(U[,2],main="Traceplot, PC2")
```



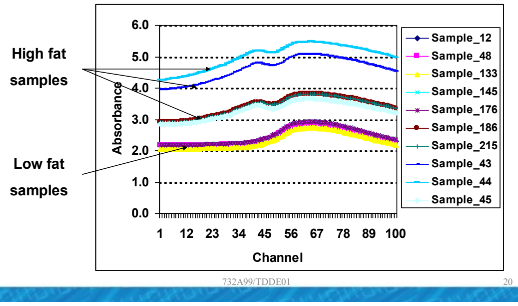


Which components contribute to PC1-2?

19

Absorbance records for ten samples of chopped meat

PCA2 captures the most of remaining variation



20

PCA for high-dimensional data

- Standard PCA for p>>N
  - At most N eigenvalues are nonzero
  - Running time is $O(p^3)$

- High-dimensional PCA
  1. Use $S' = \frac{1}{N} X X^T$ (instead of $S = \frac{1}{N} X^T X$)
  2. Eigenvalues do not change
  3. Eigenvectors of S are $X^T v_i$

21

Probabilistic PCA

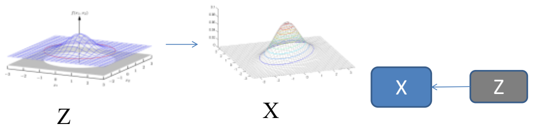
- z_i-latent variables, x_i- observed variables

$$\boldsymbol{z} \sim \boldsymbol{N}(0, \boldsymbol{I})$$
$$\boldsymbol{x}|\boldsymbol{z} \sim \boldsymbol{N}(\boldsymbol{x}|\boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu}, \sigma^2 \boldsymbol{I})$$

- Alternatively

$$\boldsymbol{z} \sim \boldsymbol{N}(0, \boldsymbol{I}), \boldsymbol{x} = \boldsymbol{\mu} + \boldsymbol{W}\boldsymbol{z} + \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \boldsymbol{N}(\boldsymbol{0}, \boldsymbol{\sigma}^2 \boldsymbol{I})$$

- **Interpretation**: Observed data (X) is obtained by rotation, scaling and translation of standard normal distribution (Z) and adding some noise.



22

Probabilistic PCA

- **Aim**: extract Z from X
- Distribution of x:

$$\boldsymbol{x} \sim \boldsymbol{N}(\boldsymbol{\mu}, \boldsymbol{C})$$
$$\boldsymbol{C} = \boldsymbol{W}\boldsymbol{W}^T + \boldsymbol{\sigma}^2 \boldsymbol{I}$$

- Rotation invariance
  - Assume that x was generated from $z' = Rz, RR^T = I,\ p(x)$ does not change!

$$\boldsymbol{x}|\boldsymbol{z}' \sim \boldsymbol{N}(\boldsymbol{x}|\boldsymbol{W}\boldsymbol{z}' + \boldsymbol{\mu}, \sigma^2 \boldsymbol{I})$$

  - **Model will not be able find latent factors uniquely!** ☹
    - It does not distinguish z from z'

23

Probabilistic PCA

- Estimation of parameters: ML

Theorem. ML estimates are given by

$$\mu_{ML} = \bar{x}$$
$$W_{ML} = U_M (L_M - \sigma^2_{ML} I)^{\frac{1}{2}} R$$
$$\sigma^2_{ML} = \frac{1}{p-M} \sum_{i=M+1}^{p} \lambda_i$$

- U_M matrix of M eigenvectors
- L_M diagonal matrix of M eigenvalues
- R any orthogonal matrix

24

Probabilistic PCA

- Estimation of Z
  - Use mean of posterior

$$\hat{z} = (W_{ML}^T W_{ML} + \sigma_{ML}^2 I)^{-1} W_{ML}^T (x - \mu)$$

- Connection to standard PCA
  - Assume $R = I, \sigma^2 = 0$ → get standard PCA components scaled by inverse root of eigenvalues

$$Z = XUL^{-\frac{1}{2}}$$

Advantages of probabilistic PCA

- More settings to specify → more flexible
- Can be faster when M<<p
- Missing values can be handled
- M can be derived if a Bayesian version is used
- Probabilistic PCA can be applied to classification problems directly
- Probabilistic PCA can generate new data

Probabilistic PCA in R

- Use **pcaMethods** from Bioconductor
- Install
  - source("https://bioconductor.org/biocLite.R")
  - biocLite("pcaMethods")

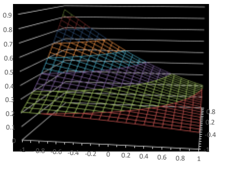
Ppca(data, nPcs,...)

Results: scores, loadings...

Independent component analysis (ICA)

- Probabilistic PCA does not capture latent factors
  - Rotation invariance
- Let's choose distribution which is not rotation invariant → will get unique latent factors
- Choose non-Gaussian $p_i(z) = p(z)$
- Assuming latent features are **independent**

$$p(z) = \prod_{i=1}^{M} p_i(z_i)$$



ICA

- Model

$$x = \mu + Wz + \epsilon, \qquad \epsilon \sim \boldsymbol{N}(0, \Sigma)$$

- **Estimation A:** **Maximum likelihood** ($V = W^{-1}$)
  - Assuming noise-free x

$$\max_V \sum_{i=1}^{n} \sum_{j=1}^{p} \log\left(p_j\left(v_j^T x_i\right)\right)$$

Subject to $||v_i|| = 1$

29

ICA

- Setting $G_j(z) = -\log\left(p_j(z)\right), z_j = v_i^T \boldsymbol{x}$ and assuming large sample

$$\min_V \sum_{j=1}^{p} E(G_j(z_j))$$

Subject to $||v_i|| = 1$

- **Prewhitening**
  - Use PCA: $X' = XU$
  - Computing z_is for given V: $\quad Z = X'V$

30

ICA

- **Estimation B**: maximize negentropy
  - ICA looks for model which is as much non-Gaussian as possible

- **Entropy** $H(z) = -\int p(z) \log p(z)\, dz = E\left(-\log p\,(z)\right)$

- **Negentropy** $J(z_i) = H(z_i') - H(z_i)$
  - $z_i' \sim N(Ez_i, var(z_i))$

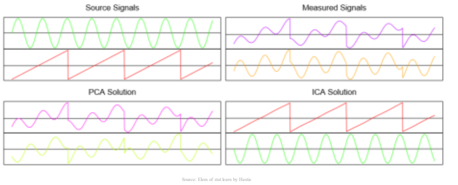
- **Negentropy maximization**

$$\max_V \sum_{j=1}^{p} J(z_j) = \min_V \sum_{j=1}^{p} H(z_j) = \min_V \sum_{j=1}^{p} E\left(-\log p\,(z_j)\right)$$

31

ICA

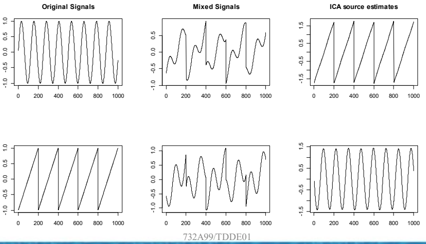
- Example



32

Independent component analysis: R

```
S <- cbind(sin((1:1000)/20), rep((((1:200)-100)/100), 5)) #Generate data
A <- matrix(c(0.291, 0.6557, -0.5439, 0.5572), 2, 2)
X <- S %*% A
a <- fastICA(X, 2, alg.typ = "parallel", fun = "logcosh", alpha = 1,
     method = "R", row.norm = FALSE, maxit = 200, tol = 0.0001, verbose = TRUE) #ICA
```



33