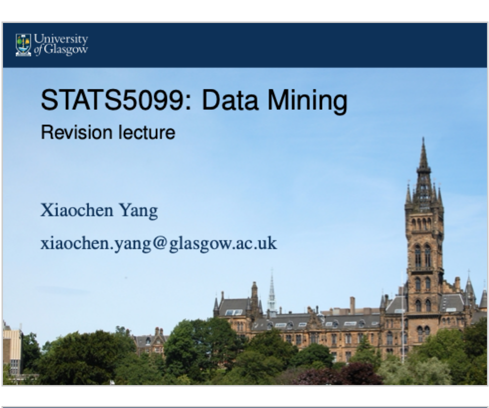
rev

Thursday, July 22, 2021 8:56 AM

Revision lecture...



University of Glasgow

STATS5099: Data Mining

Revision lecture

Xiaochen Yang

xiaochen.yang@glasgow.ac.uk

Data mining methods

Dimension reduction
- Principal component analysis (weeks 1-2)
- Multidimensional scaling (week 2)

Classification
- *k*-nearest neighbours (week 3)
- Linear discriminant analysis (week 3)
- Classification trees, bagging, random forest (week 4)
- Support vector machines (week 5)
- Neural networks (week 6)

Clustering
- Agglomerative hierarchical clustering (week 8)
- *K*-means clustering (week 9)
- *K*-medoids clustering (week 9)

Recommendation systems
- Content-based filtering (week 10)
- Collaborative filtering (week 10)

2/20

Revision suggestions

- For each method,
  - describe the method
  - explain the procedure of the method, e.g. by using a simple example
  - understand how to apply it in practice, e.g. any parameters
  - understand its applicability, strength and/or limitation
  - implement in R and interpret the results
- Across variants of one method / methods in the same category,
  - understand when to use one or the other

3/20

Principal component analysis

- describe the method
  - PCA is a linear dimensionality reduction method.
  - It finds a small number of uncorrelated linear combinations of the original variables which explain most of the variation in the data.

4/20

Principal component analysis

- explain the procedure of the method
  - The first principal component is the linear combination of original variables with the maximum variance, subject to the normalising constraint:

$$\text{first PC} = \text{linear combination } Y_1 \text{ that maximises } \mathrm{Var}(Y_1)$$
$$\text{subject to } \boldsymbol{a}_1^T\boldsymbol{a}_1 = 1$$

  - The j^{th} PC is calculated in the same way (i.e. with the objective of maximising variance), with the condition that it is uncorrelated with previous PCs

$$j\text{th PC} = \text{linear combination } Y_j \text{ that maximises } \mathrm{Var}(Y_j)$$
$$\text{subject to } \boldsymbol{a}_j^T\boldsymbol{a}_j = 1 \text{ and } \mathrm{Cov}(Y_j, Y_k) = 0 \text{ for } k < j$$

5/20

Principal component analysis

- explain the procedure of the method
  - The first PC is the eigenvector associated with the largest eigenvalue of the covariance matrix Σ.
  - The j^{th} PC is the eigenvector associated with the j^{th} largest eigenvalue.

5/20

Principal component analysis

- understand how to apply it in practice
  - sample covariance matrix or sample correlation matrix?
  - how many PCs?
    - proportion of Variation
    - Cattell's method
    - Kaiser's method

6/20

Principal component analysis

- understand its applicability, strength and limitation
  - *linear* dimension reduction for *continuous* data
  - particularly useful when the original variables are highly correlated
  - no need to make assumptions on model or data distribution
  - has an analytical solution
  - can be severely distorted by outliers (first PC determined by outliers)

7/20

Principal component analysis

- implement in R and interpret the results
  - `princomp(data, cor=T)`
  - `summary(pca.result)`

```
Importance of components:
                          Comp.1    Comp.2    Comp.3     Comp.4     Comp.5
Standard deviation     2.1861050 1.5899422 1.1601299 0.96170782 0.92825842
Proportion of Variance 0.3676196 0.1944551 0.1035309 0.07114476 0.06628182
Cumulative Proportion  0.3676196 0.5620747 0.6656056 0.73675038 0.80303220
                          Comp.6     Comp.7     Comp.8     Comp.9    Comp.10
Standard deviation     0.8004340 0.74398164 0.57942382 0.53978997 0.50243903
Proportion of Variance 0.0492842 0.04257759 0.02582554 0.02241332 0.01941884
Cumulative Proportion  0.8523164 0.89489400 0.92071953 0.94313286 0.96255170
                          Comp.11    Comp.12     Comp.13
Standard deviation     0.48133977 0.40737507 0.298639313
Proportion of Variance 0.01782215 0.01276573 0.006860418
Cumulative Proportion  0.98037386 0.99313958 1.000000000
```

8/20

Principal component analysis

- implement in R and interpret the results
  - `princomp(data, cor=T)`
  - `summary(pca.result)`
  - `pca.result$loadings`

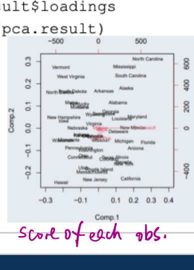
```
wine.pca$loadings[,1:8]

                  Comp.1  Comp.2 Comp.3  Comp.4 Comp.5  Comp.6 Comp.7  Comp.8
Alcohol           0.1491  0.4856  0.174  0.0093  0.266  0.2068  0.061  0.4613
Malic.acid       -0.2425  0.2234 -0.088 -0.5407 -0.056  0.5162 -0.438  0.0905
Ash              -0.0096  0.3283 -0.640  0.1946  0.139  0.1418  0.143 -0.2809
Alcalinity.of.ash -0.2503 -0.0081 -0.612 -0.0838 -0.084 -0.1255  0.310  0.4339
Magnesium         0.1387  0.2991 -0.112  0.3673 -0.722  0.0275 -0.326 -0.0955
Total.phenols     0.3918  0.0639 -0.137 -0.2055  0.139 -0.0855  0.020 -0.3900
Flavanoids        0.4267 -0.0054 -0.112 -0.1577  0.107 -0.0064  0.046 -0.1290
Nonfl.phenols    -0.2989  0.0304 -0.157  0.1833  0.500 -0.2771 -0.600 -0.1735
Proanthocyanins   0.3100  0.0373 -0.153 -0.3998 -0.158 -0.5666 -0.331  0.2865
Colour.intensity -0.0876  0.5247  0.179 -0.0664  0.079 -0.3992  0.233 -0.0033
Hue               0.2939 -0.2753 -0.130  0.4192  0.178  0.0810 -0.221  0.4653
OD280.OD315       0.3731 -0.1650 -0.167 -0.1886  0.092  0.2671  0.030 -0.0400
Proline           0.2876  0.3636  0.096  0.2272  0.161  0.1081 -0.074  0.0607
```

8/20

Principal component analysis

- implement in R and interpret the results
  - `princomp(data, cor=T)`
  - `summary(pca.result)`
  - `pca.result$loadings`
  - `biplot(pca.result)`



score of each obs.

8/20

PCA *vs* MDS

	PCA	metric MDS	nonmetric MDS
data type	continuous variables	proximity data (similarity or dissimilarity matrix)	
data structure	data lies in a linear subspace	–	–
objective	maximise variance	preserve pairwise distance	
computation	analytical solution (eigen vectors)	gradient descent iterative	iterative (longer time to converge)
solution	unique (up to sign flip)	sensitive to initial configurations different pts, differet solution	

9/20

Demo exam Q2

(a) To reduce the dimension of this data set, a researcher has applied principal component analysis (PCA) based on the correlation matrix. Suggest why PCA might have been run on the correlation matrix instead of the covariance matrix? **[2 MARKS]**

10/20

Demo exam Q2

(b) Partial output from the principal component analysis is given below.

```
> glass.pca <- princomp(Glass,cor=TRUE)
> glass.pca
Standard deviations:
Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9
  1.58   1.43   1.19   1.08   0.96   0.73   0.61   0.25   0.04
> glass.pca$loadings
Loadings:
   Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9
RI  0.545  0.286         0.147         0.115         0.752
Na -0.258  0.270 -0.385  0.491 -0.154 -0.558  0.149  0.128 -0.312
Mg  0.111 -0.594         0.379 -0.124  0.308 -0.206        -0.577
Al -0.429  0.295  0.329 -0.138               -0.699  0.274 -0.192
Si -0.229 -0.155 -0.459 -0.653                0.216  0.380 -0.298
K  -0.219 -0.154  0.663         0.307 -0.244  0.504  0.110 -0.261
Ca  0.492  0.345        -0.276  0.188 -0.149        -0.399 -0.579
Ba -0.250  0.485         0.133 -0.251  0.657  0.352 -0.145 -0.198
Fe  0.186         0.284 -0.230 -0.873 -0.243
```

Comment on the loadings of the first principal component. Comment on the role of the variable Fe in the principal component analysis. **[4 MARKS]**

10/20

Demo exam Q2

(c) The researcher chose to use the Proportion of Variation approach to determine the number of principal components to be retained. Based on the previous R output, decide how many components should be kept in order to explain 85% of the variability of the data set. **[4 MARKS]**

10/20

Support vector machines

- describe the method
  - A classification method
  - The method directly tries to fit decision boundaries in such a way as to maximise the margin or separation between the two classes.
  - Three types: hard-margin linear SVM, soft-margin SVM, nonlinear SVM

11/20

Support vector machines

- explain the procedure of the method (hard-margin SVM)
  - choose the *hyperplane* parameters to maximise the *margin*
  - solve the optimisation problem:

$$\min_{b,\boldsymbol{w}} \|\boldsymbol{w}\|$$
$$\text{subject to } g_i(\boldsymbol{w}^\top\boldsymbol{x} + b) \geq 1 \;\; \forall i = 1, \ldots, n$$

12/20

Support vector machines

- understand how to apply it in practice
  - scale the data before applying SVM

soft-margin SVM
  - cost parameter increase / decrease, result...

nonlinear SVM
  - kernel function parameters in kernel, evaluate the effect

13/20

Support vector machines

- understand its applicability, strength and/or limitation
  - designed for binary classification (one vs all, one vs one)
  - no model or data assumption
  - very flexible and suitable for nonlinear non-separable data
  - fast inference (knn quite long)
  - sensitive to hyperparameters Cost para., para in kernel
  - no means of assessing uncertainty
  - low interpretability

14/20

Support vector machines

- implement in R and interpret the results
  - `svm(Y~., data, type="C-classification", cost = C, kernel="radial", ...)`

soft margin

```
Call:
svm(formula = sp ~ ., data = train.data, kernel = "linear",
    cost = C.sel, type = "C-classification")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  0.5

Number of Support Vectors:  55
 ( 28 27 )


Number of Classes:  2

Levels:
 B O
```

kernel

parameter

class 1 class 2

15/20

Support vector machines

- implement in R and interpret the results
  - `svm(Y~., data, type="C-classification", cost = C, kernel="radial", ...)`
  - `tune.svm` or `tune` tune the paras in SVM

```
tune.svm(sp~.,data=train.data,type="C-classification",kernel="linear",
         cost=C.val)
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 cost
  0.5

- best performance: 0
```

vector of all possible parameters

15/20

Demo exam Q2

See the demo exam.

16/20

Q&As

- weekly material tasks and tutorial question, which is similar to exam style question
- If we wanted to practice some more exam style questions, would there be another module (past or present) that we can search the past papers portal for, which would provide us with these?

  A: multivariate methods (NB: Topics and lecture materials are different)
- an additional drop-in Q&A session at 12pm next Wednesday (28th July)

17/20

Q&As

- LDA
- biplot
- Tutorial sheet 1 Q2
- Tutorial sheet 1 Q3

18/20

Q&As: LDA

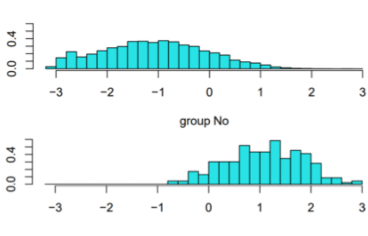
- Q: The notes say we have to evaluate each group to see which one has the largest LDF, but there is only one LDF for two groups, so what are we comparing the result of the equation with in order to choose which group to assign the data to?
- A: We will assign the object $\boldsymbol{x}$ to the class with the largest linear discriminant function (LDF)

$$LDF_g(\boldsymbol{x}) = \boldsymbol{x}^\top\Sigma^{-1}\boldsymbol{\mu}_g - \frac{1}{2}\boldsymbol{\mu}_g^\top\Sigma^{-1}\boldsymbol{\mu}_g + \log\pi_g$$

19/20

Q&As: LDA

```
Coefficients of linear discriminants:
                 LD1
balance 2.218221e-03
income  1.056657e-05
```

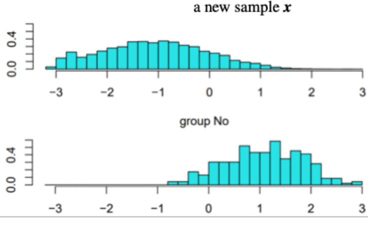


19/20

Q&As: LDA

```
Coefficients of linear discriminants:
                 LD1
balance 2.218221e-03
income  1.056657e-05
```

1. create a classification rule, e.g. $\text{LD1} < 0 \Rightarrow \text{group No}$
2. calculate the value of LD1 for a new sample $\boldsymbol{x}$



19/20

Q&As: Tutorial 1 Q3

- Show that principal components for standardised variables can be obtained from the eigenvectors of the correlation matrix $\boldsymbol{R}$ of the random vector $\boldsymbol{X}$.
- Q: The model answer proves that Cov(Z) = R, but I don't understand how that answers the question, which is about the PCs coming from the eigenvalues
- A: PCs for standard variables are eigenvectors of the covariance matrix of the standard variables, which equals to the correlation matrix $\boldsymbol{R}$ of the original variables $\boldsymbol{X}$

20/20

Q&As: Tutorial 1 Q3

- Show that principal components for standardised variables can be obtained from the eigenvectors of the correlation matrix $\boldsymbol{R}$ of the random vector $\boldsymbol{X}$.

Let's standardise original variables $\boldsymbol{X} = (X_1, \ldots, X_p)^T$ to $\boldsymbol{Z} = (Z_1, \ldots, Z_p)^T$, where

$$Z_i = \frac{X_i - \mu_i}{\sqrt{\sigma_{ii}}}$$

Using a diagonal matrix $\boldsymbol{V}^{1/2}$ for standard deviations $\sqrt{\sigma_{ii}}$, we can write the standardisation in matrix notation

$$\boldsymbol{Z} = (\boldsymbol{V}^{1/2})^{-1}(\boldsymbol{X} - \boldsymbol{\mu})$$

Clearly, $\mathbb{E}(\boldsymbol{Z}) = 0$.

$$\mathrm{Cov}(\boldsymbol{Z}) = (\boldsymbol{V}^{1/2})^{-1}\mathrm{Cov}(\boldsymbol{X})(\boldsymbol{V}^{1/2})^{-1} = (\boldsymbol{V}^{1/2})^{-1}\Sigma(\boldsymbol{V}^{1/2})^{-1} = \boldsymbol{R}$$

20/20