

# Biometry. Lecture 24

Alexey Shipunov

Minot State University

May 7, 2014



# Outline

- 1 Questions and answers
  - Previous final question

- 2 Multivariate statistics, or Data Mining

- Generic methods
- Principal Component Analysis (PCA)
- Correspondence analysis
- Similarity
- Multi-dimensional scaling
- Cluster analysis
- Classification (machine learning)
- Linear Discriminant Analysis (LDA)
- Regression trees (recursive partitioning)
- Advanced methods of classification



## 1 Questions and answers

- Previous final question

## 2 Multivariate statistics, or Data Mining

- Generic methods
- Principal Component Analysis (PCA)
- Correspondence analysis
- Similarity
- Multi-dimensional scaling
- Cluster analysis
- Classification (machine learning)
- Linear Discriminant Analysis (LDA)
- Regression trees (recursive partitioning)
- Advanced methods of classification



```
> setwd("<working folder>")  
or  
"Change dir"  
in menu!
```



# Questions and answers

## Previous final question



# Previous final question: the answer

Why do biologists use multivariate methods?



# Previous final question: the answer

Why do biologists use multivariate methods?

- To see the structure



# Multivariate statistics, or Data Mining

## Generic methods





# Matrix graph: correction

```
> pairs(iris[1:4], pch=21, bg=(1:3)[iris$Species])
```



# Parallel coordinates plot

```
> eq8 <- read.table("http://ashipunov.info/data/eq8.txt",  
+ h=T)  
> library(MASS)  
> parcoord(eq8[,-1], col=rep(rainbow(8), table(eq8[,1])))  
> legend("top", names(table(eq8[,1])), fill=rainbow(8),  
+ ncol=4)
```



# Multivariate statistics, or Data Mining

## Principal Component Analysis (PCA)



# Principal Component Analysis

- Principal Component Analysis tries to achieve the best projection of multivariate cloud, taking into account as many characters (dimensions) as possible
- All characters are transformed into components; first component is the most important, second and third are also significant



# PCA for `iris` data

```
> iris.pca <- princomp(scale(iris[,1:4]))  
> plot(iris.pca, main="") # this is technical screeplot  
> iris.p <- predict(iris.pca)  
> plot(iris.p[,1:2], type="n", xlab="PC1", ylab="PC2")  
> text(iris.p[,1:2], labels=abbreviate(iris[,5], 1,  
+ method="both"))  
> loadings(iris.pca)
```



# Inferential PCA (library `ade4`)

```
> library(ade4)
> iris.d <- dudi.pca(iris[,1:4], scannf=FALSE)
> s.class(iris.d$li, iris[,5])
> randtest(bca(iris.d, iris[,5], scannf=FALSE))
```



# Multivariate statistics, or Data Mining

## Correspondence analysis



# Correspondence analysis

- You may think about PCA as a multivariate derivative of correlation analysis, and correspondence analysis may be imagined as a derivative of contingency tables analysis
- Unique feature of correspondence analysis is an ability to show **both** rows and columns on one graph





# Simple example of correspondence visualization

```
> library(MASS)
> caith
> biplot(corresp(caith, nf=2))
```

`caith` is the embedded data on the cross-classification of people in Caithness, Scotland, by eye and hair colour

Library `vegan` contains more advanced methods and graphs, represented in particular by functions `cca()` and `decorana()`



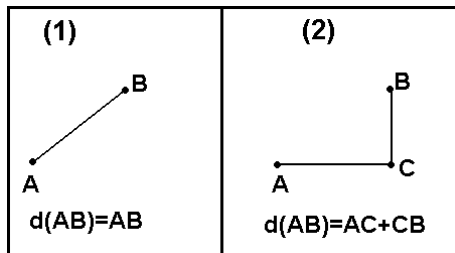
# Multivariate statistics, or Data Mining

## Similarity



# Distance and similarity

- Distance is simple a numerical measure of similarity
- Euclidean distance is (1) hypotenuse; manhattan distance (2) is a sum of legs



# Closeness and distance

```
> ma <- data.frame(V1=c(7,3,5), V2=c(7,5,3))  
> row.names(ma) <- c("A", "B", "C")  
> dist(ma) # Euclidean is default  
> dist(ma, method="manhattan")  
> iris.d <- dist(iris[,1:4])  
> library(cluster)  
> iris.dist <- daisy(iris[,1:4], metric="manhattan")
```

`daisy()` function is more universal since it can work with both binary and measurement variables.



# Multivariate statistics, or Data Mining

## Multi-dimensional scaling



# Multi-dimensional scaling

- Multi-dimensional scaling may be seen as making a geographic map from all pairs of distances
- Results are often similar to PCA but axes are not connected with any particular character



# Scaling examples

```
> example(cmdscale)
> eurodist
> iris.c <- cmdscale(iris.dist)
> plot(iris.c[,1:2], type="n", xlab="Dim. 1",
+ ylab="Dim. 2")
> text(iris.c[,1:2], labels=abbreviate(iris[,5], 1,
+ method="both.sides"))
```



# Multivariate statistics, or Data Mining

## Cluster analysis





# Cluster analysis

- Clusterization is the making groups
- Hierarchical clusterization makes trees (dendrograms)



# Hierarchical clustering

```
> plot(hclust(dist(ma)))  
# We will choose every fifth row  
> iriss <- iris[seq(1,nrow(iris), 5),]  
> iriss.dist <- daisy(iriss[, 1:4])  
> iriss.h <- hclust(iriss.dist, method="ward")  
> plot(iriss.h, labels=abbreviate(iriss[,5], 1,  
+ method="both.sides"))
```



# Support for branches

```
> library(pvclust)
> irisst <- t(iriss[, 1:4])
> colnames(irisst) <- paste(abbreviate(iriss[,5], 3),
+ colnames(irisst))
> iriss.pv <- pvclust(irisst, method.dist="manhattan",
+ method.hclust="ward", nboot=100)
> plot(iriss.pv, col.pv=c(1, 0, 0))
```



# Another hierarchical clustering example (very simple)

```
> fences <- read.table(  
+ "http://ashipunov.info/data/fences.txt", h=T)  
> library(cluster)  
> str(fences)  
> fences.d <- daisy(fences)  
> summary(fences.d)  
> plot(hclust(fences.d))
```



# Fuzzy clustering

```
> iris.f <- fanny(iris[,1:4], 3)
> plot(iris.f, which=1, main="")
> head(data.frame(sp=iris[,5], iris.f$membership))
```



# Multivariate statistics, or Data Mining

## Classification (machine learning)



# Classification (machine learning)

- Machine learning, or classification is always based on the example where objects are already distributed into groups
- These methods are trying to find a best classification algorithm



# Multivariate statistics, or Data Mining

## Linear Discriminant Analysis (LDA)





# Linear Discriminant Analysis (LDA)

- Linear discriminant analysis is based on the idea that classification could be made on a bases of linear equations
- This is a parametric method



# LDA example

```
> library(MASS)
> iris.train <- iris[seq(1,nrow(iris),5),]
> iris.unknown <- iris[-seq(1,nrow(iris),5),]
> iris.lda <- lda(Species ~ . , data=iris.train)
> iris.ldap <- predict(iris.lda, iris.unknown[,1:4])$class
> table(iris.ldap, iris.unknown[,5])
```



# LDA testing

```
> ldam <- manova(as.matrix(iris.unknown[,1:4]) ~  
+ iris.ldap)  
> summary(ldam, test="Wilks")
```

“Wilks” value is not only a statistic, it is also a likelihood ratio: for better classifications, Wilks is closer to 0



# LDA visualization

```
> iris.lda2 <- lda(scale(iris[,1:4]), iris[,5])  
> iris.ldap2 <- predict(iris.lda2, dimen=2)$x  
> plot(iris.ldap2, type="n", xlab="LD1", ylab="LD2")  
> text(iris.ldap2, labels=abbreviate(iris[,5], 1,  
+ method="both.sides"))
```



# Multivariate statistics, or Data Mining

## Regression trees (recursive partitioning)



# Regression trees (recursive partitioning)

- Regression trees, or recursive partitioning are based on the same idea as biological descriptive keys
- On each step, methods searches for the best separation between members of group



# Regression tree example I

```
> library(tree)
> iris.tree <- tree(Species ~ ., data=iris)
> plot(iris.tree); text(iris.tree)
```



# Regression tree example II

```
> eq <- read.table("http://ashipunov.info/data/eq.txt", h=TRUE)
> eq.tree <- tree(SPECIES ~ ., data=eq)
> plot(eq.tree); text(eq.tree)
```





# Multivariate statistics, or Data Mining

## Advanced methods of classification



# Advanced methods of classification

- “Random Forest” is based on the construction of multiple regression trees
- “Support Vector Machines” try to find a hyperplane which separates objects best



# Random Forest example

```
> library(randomForest)
> set.seed(17)
> iris.rf <- randomForest(Species ~ ., data=iris.train)
> iris.rfp <- predict(iris.rf, iris.unknown[,1:4])
> table(iris.rfp, iris.unknown[,5])
```



# Random Forest visualization

```
> set.seed(17)
> iris.urf <- randomForest(iris[,1:4])
> MDSplot(iris.urf, iris[,5])
```



# SVM example

```
> library(e1071)
> iris.svm <- svm(Species ~ ., data=iris.train)
> iris.svmp <- predict(iris.svm, iris.unknown[,1:4])
> table(iris.svmp, iris.unknown[,5])
```



# Finishing...

```
> savehistory("20140507.r")
```



# Short anonymous absolutely voluntary survey

- 1 What do you **like** most in biometrics course?
- 2 What do you **dislike** most in biometrics course?
- 3 **Which lab** do you remember most of all?
- 4 Please grade (1—bad, 5—excellent):
  - 1 Lectures
  - 2 Labs
  - 3 Final questions
  - 4 Exams
- 5 Please recommend something for the Spring 2015 Biometrics.

