Лабораторная работа № 4: Классификация и кластеризация временных рядов в R

### 1. Цель работы

* Освоить основные методы классификации и кластеризации
* Приобрести основные навыки работы с классификацией и кластеризацией временных рядов в R

### 2. Задачи

* Углубить и закрепить знания по основным методам классификации и кластеризации.
* Научить делать классификацию и кластеризацию временных рядов в R
* Совершенствовать навыки самостоятельной работы.

### 3. План выполнения работы

### 3.1 Основные алгоритмы классификации и кластеризации

* Алгоритмы классификации:

Linear classifiers

Logistic regression

Naive Bayes classifier

Fisher’s linear discriminant

Support vector machines

Least squares support vector machines

Quadratic classifiers

Kernel estimation

k-nearest neighbor

Decision trees

Random forests

Neural networks

Learning vector quantization

* Алгоритмы кластеризации:

Linear clustering algorithm

k-means clustering algorithm

Fuzzy c-means clustering algorithm

Hierarchical clustering algorithm

Gaussian(EM) clustering algorithm

Quality threshold clustering algorithm

Non-linear clustering algorithm

MST based clustering algorithm

kernel k-means clustering algorithm

Density-based clustering algorithm

### 3.2 Класстеризация временных рядов

* Загрузка данных для прогнозирования (в качестве примера воспользуемся данными “synthetic\_control.data”)
* Уставовим необходимый пакет: dtw

Например:

# load data  
sc <- read.table("synthetic\_control.data", header = F, sep = "")  
# randomly sampled n cases from each class, to make it easy for plotting  
n <- 10  
s <- sample(1:100, n)  
idx <- c(s, 100+s, 200+s, 300+s, 400+s, 500+s)  
sample2 <- sc[idx,]  
observedLabels <- c(rep(1,n), rep(2,n), rep(3,n), rep(4,n), rep(5,n), rep(6,n))  
  
# compute DTW distances  
  
library(dtw)

## Loading required package: proxy

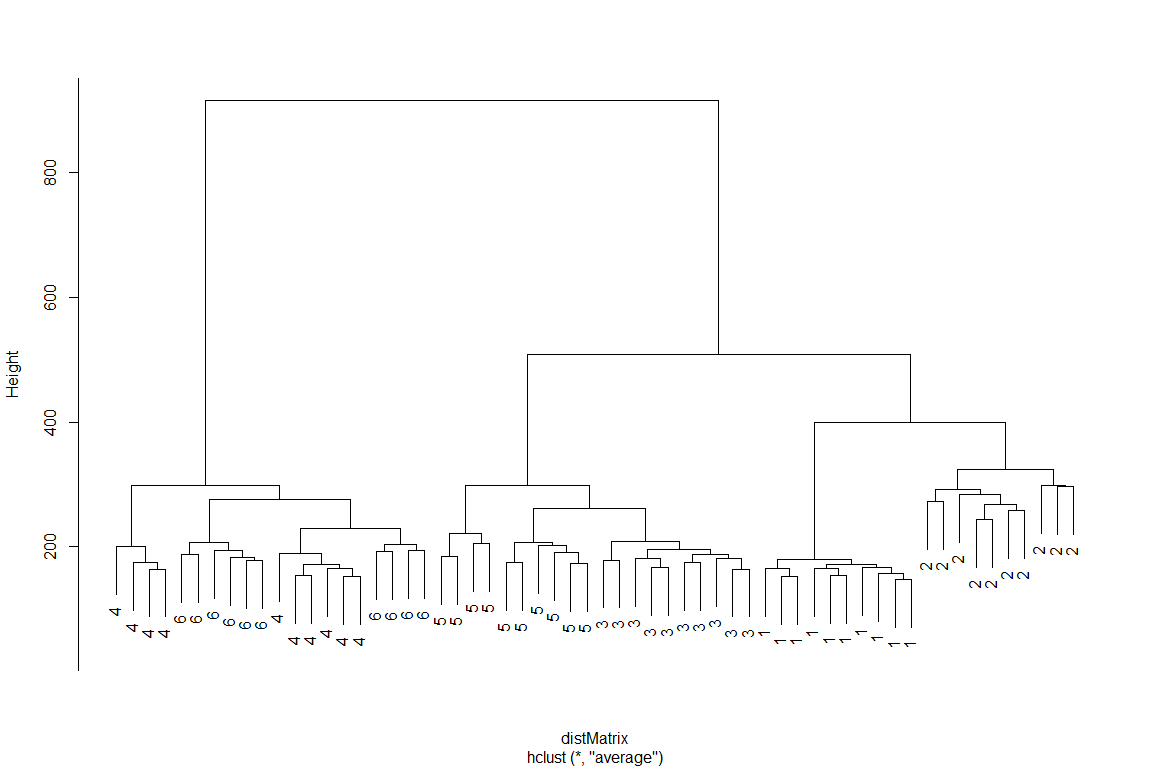
##   
## Attaching package: 'proxy'

## The following objects are masked from 'package:stats':  
##   
## as.dist, dist

## The following object is masked from 'package:base':  
##   
## as.matrix

## Loaded dtw v1.18-1. See ?dtw for help, citation("dtw") for use in publication.

distMatrix <- dist(sample2, method="DTW")  
  
# hierarchical clustering  
hc <- hclust(distMatrix, method= "average")  
plot(hc, labels=observedLabels, main = "")



### 3.3 Классификация временных рядов

* Загрузка данных для прогнозирования (в качестве примера воспользуемся данными “synthetic\_control.data”)
* Уставовим необходимые пакеты: wavelets, party

Например:

# load data  
sc <- read.table("synthetic\_control.data", header = F, sep = "")  
# extracting DWT coefficients (with Haar filter)  
library(wavelets)  
wtData <- NULL  
  
for (i in 1:nrow(sc)) {  
 a <- t(sc[i,])  
 wt <- dwt(a, filter="haar", boundary="periodic")  
 wtData <- rbind(wtData, unlist(c(wt@W,wt@V[[wt@level]])))  
 }  
wtData <- as.data.frame(wtData)  
  
# set class labels into categorical values  
classId <- c(rep("1",100), rep("2",100), rep("3",100) ,rep("4",100), rep("5",100), rep("6",100))  
wtSc <- data.frame(cbind(classId, wtData))  
  
# build a decision tree with ctree() in package party  
library(party)

## Loading required package: grid

## Loading required package: mvtnorm

## Loading required package: modeltools

## Loading required package: stats4

## Loading required package: strucchange

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

## Loading required package: sandwich

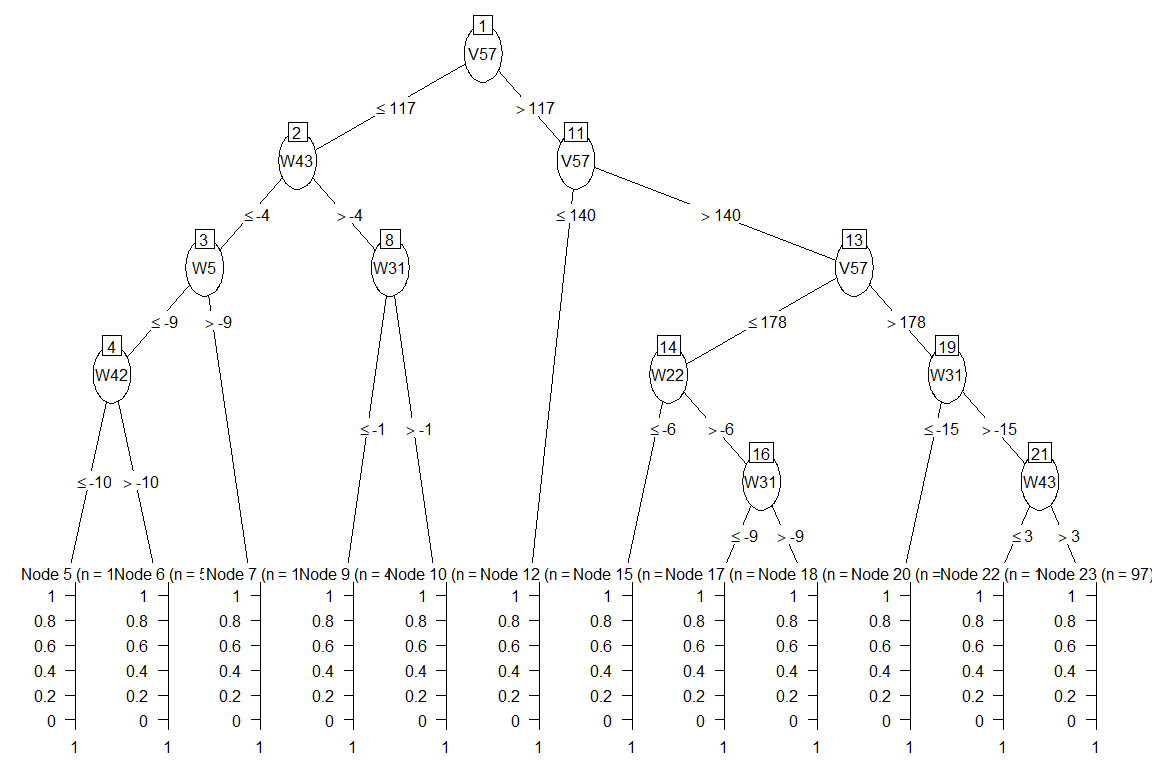
ct <- ctree(classId ~ ., data=wtSc, controls = ctree\_control(minsplit=30, minbucket=10, maxdepth=5))  
pClassId <- predict(ct)  
# check predicted classes against original class labels  
table(classId, pClassId)

## pClassId  
## classId 1 2 3 4 5 6  
## 1 97 3 0 0 0 0  
## 2 1 99 0 0 0 0  
## 3 0 0 81 0 19 0  
## 4 0 0 0 63 0 37  
## 5 0 0 16 0 84 0  
## 6 0 0 0 1 0 99

# accuracy  
(sum(classId==pClassId)) / nrow(wtSc)

## [1] 0.8716667

plot(ct, ip\_args=list(pval=FALSE), ep\_args=list(digits=0))



### 4. Задание:

**Задание № 1**: Выполнение классификации для набора данных Iris.