R - příklady

Jan Knotek

I. Play or Not To Play

Outlook	Temp.	Humidity	Windy	Play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

TPONTPNom.csv – textový formát, data oddělena čárkami

1. Načtení CSV dat do R

Načtení:

```
> data <- read.csv(file.choose())
```

Zobrazení:

```
> data
```

První řádek:

```
> data[1,]
```

3-5 sloupec:

```
> data[,3:5]
```

1. Načtení dat do R

Další možnosti:

- read.table – obecnější, více nastavení
- read.csv2 – pro použití s daty, kde se používá desetinná čárka místo tečky

Package *tm* má vlastní systém (hlavně pro text):

- readPlain, readPDF, readDOC
- getReaders()

2. Rozhodovací strom

```
> library(rpart)
> tree<-rpart(Play~., data, method="class")
> tree
n= 14
node), split, n, loss, yval, (yprob)
      * denotes terminal node
1) root 14 5 yes (0.3571429 0.6428571) *
```

Vznikl jen jeden list – 64,29% šance správné klasifikace, pokud klasifikujeme vždy „yes“

Málo dat, proto je třeba upravit parametry rpart

2. Rozhodovací strom

```
> tree<-rpart(Play~., data, method="class",
  control=rpart.control(minsplit=5))
> tree
n= 14
node), split, n, loss, yval, (yprob)
      * denotes terminal node
1) root 14 5 yes (0.3571429 0.6428571)
  2) Outlook=rainy,sunny 10 5 no (0.5000000 0.5000000)
    4) Humidity=high 5 1 no (0.8000000 0.2000000) *
    5) Humidity=normal 5 1 yes (0.2000000 0.8000000) *
  3) Outlook=overcast 4 0 yes (0.0000000 1.0000000) *
```

Parametr minsplit – minimum instancí v uzlu, kdy se můžeme pokusit o jeho rozdělení

3. Predikce pro trénovací data

Zavolání funkce predict pro model „tree“ na trénovacích datech – vybrány první 4 atributy bez sloupce „Play“, který se snažíme predikovat:

```
> tree.predictions <- predict(tree, data[,1:4], type="class")
> table(data[,5], tree.predictions)
 tree.predictions
     no yes
  no   4   1
  yes  1   8
```

Testem na trénovacích datech zjistíme pouze, jestli model není úplně špatně – nelze vyvozovat žádné jiné závěry!!!

4. Cross-validation

```
err.vect <- vector()
for(j in 1:10) {    # 10 pokusů
   select <- sample(1:nrow(data), 0.9*nrow(data))
                                    # náhodná permutace dat
   train <- data[select,]           # 90% dat pro trénink
   test <- data[-select,]           # zbylá data pro test (10%)
   tree <- rpart(Play~., train, control=rpart.control(minsplit=5))
   pred <- predict(tree, test[,1:4], type="class")
   cmx<-table(test[, "Play"], pred)  # sloupec Play pro ověření predikce
   err<- 1 - ( sum(diag(cmx)) / sum(cmx) )
   err.vect <- c(err.vect, err)
}
err.vect; mean(err.vect)
```

Error rate – poměr špatně klasifikovaných instancí ke všem instancím (z 10 error rate se udělá vektor, pak se zpočítá průměr)

II. Iris – poznávání druhů rostlin

Iris.arff – attribute relation file format (Weka):

```
@RELATION iris
@ATTRIBUTE sepallength       REAL
@ATTRIBUTE sepalwidth        REAL
@ATTRIBUTE petallength       REAL
@ATTRIBUTE petalwidth        REAL
@ATTRIBUTE class     {Iris-setosa,Iris-versicolor,Iris-virginica}
@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
...
```

1. Načtení ARFF dat do R

Načtení:

```
> library(foreign)
> data.iris <- read.arff(file.choose())
```

Zobrazení:

```
> data.iris
```

2. Rozhodovací strom

```
> library(rpart)
> tree.iris<-rpart(class~., data.iris, method="class")
> tree.iris
n= 150
node), split, n, loss, yval, (yprob)
      * denotes terminal node
1) root 150 100 Iris-setosa (0.33333333 0.33333333 0.33333333)
  2) petallength< 2.45 50   0 Iris-setosa (1.000 0.000 0.000) *
  3) petallength>=2.45 100  50 Iris-versicolor (0.0 0.500 0.500)
    6) petalwidth< 1.75 54   5 Iris-versicolor (0.0 0.907407 0.092593) *
    7) petalwidth>=1.75 46   1 Iris-virginica (0.0 0.021739 0.9782609) *
```

3. Prověření testovacími daty

Rozdělení na trénovací a testovací data:

```
> idx<-sample(150,150)     # náhodná permutace délky
                             150, do hodnoty 150
> train.iris <- data.iris[idx[1:100],]   # 2/3 dat použijeme
                                           jako trénovací data
> test.iris <- data.iris[idx[101:150],] #zbylá 1/3 dat pro test
> tree.iris <- rpart(class~., train.iris)
> pred.iris <- predict(tree.iris, test.iris[,1:4], type="class")
                # v testovacích datech opět vypustíme třídu a
                získáme predikce klasifikátoru pro testovací
                data
```

3. Prověření testovacími daty

```
> cmx.iris <- table(test.iris[, 5], pred.iris)
> cmx.iris                                  # "matice zmatení"
 pred
                 Iris-setosa Iris-versicolor Iris-virginica
  Iris-setosa             19              0              0
  Iris-versicolor          0             16              2
  Iris-virginica           0              3             10
> err.iris <- 1 - (sum(diag(cmx.iris)) / sum(cmx.iris) )
> err.iris
[1] 0.1
```

Error rate – poměr špatně klasifikovaných instancí ke všem instancím