

Číslo 5 žije a chce vědět víc  
aneb  
Úvod do machine learning v R  
spíše v pojetí biomedicíny

---

Statistické dýchánky  
na VŠE

Ondřej Klempíř  
Lubomír Štěpánek

Katedra biomedicínské informatiky  
Fakulta biomedicínského inženýrství  
České vysoké učení technické v Praze

8. listopadu 2016

# Rychlý úvod

- machine learning (*strojové učení*) je velká množina algoritmů a technik, které umožňují počítačovému systému se samostatně učit (měnit jeho vnitřní stav, aniž by právě k tomu byl explicitně naprogramován)
- algoritmy
  - ▶ s učitelem
  - ▶ bez učitele
  - ▶ kombinace obou předchozích a další
- základní typy úloh
  - ▶ regresní úloha
  - ▶ klasifikační úloha
  - ▶ shlukování

## Některé algoritmy v rámci klasifikační úlohy

# Bayesovská naivní klasifikace

- relativně jednoduchý algoritmus
- MAP princip (Maximum-A-Posteriori-Probability) - prvek je zařazen do třídy, která má na konci nejvyšší pravděpodobnost
- buď  $x = (x_1, x_2, \dots, x_m, c_i)$  jedno z pozorování v testovací množině popsané  $m$  atributy a třídou  $c_i$  pro jedno pevné  $i \in \{1, 2, \dots, k\}$
- pak pravděpodobnost, že bude  $x$  správně zařazeno do své třídy  $c_i$ , je

$$p(c_i | x) = \frac{p(x | c_i)p(c_i)}{p(x)}$$

- protože  $p(x) = \frac{1}{|\text{trénovací množina}|} = \textit{konst.}$  a pro daný dataset i  $p(c_i) = \frac{|\{y: y \text{ je třídy } c_i\}|}{|\text{trénovací množina}|} = \textit{konst.}$ , je

$$p(c_i | x) \propto p(x | c_i)$$

# Bayesovská naivní klasifikace

- za předpokladu nezávislosti atributů je

$$p(x \mid c_i) \propto \prod_{j=1}^n p(x_j \mid c_i),$$

kde  $n = |\text{testovací množina}|$ , u nespojitých atributů odhadneme

$$p(x_j \mid c_i) = \frac{|\{y: y \in \text{trén. množ.} \wedge j\text{-tý atribut } y \text{ je } x_j \wedge \text{třída } y \text{ je } c_i\}|}{|\{z: z \in \text{trénovací množina} \wedge \text{třída } z \text{ je } c_i\}|},$$

u spojitých použijeme fitting normálním rozložením a  $\phi(x_j \mid c_i)$

- $x$  je třídy  $c_i$  tak, že

$$i = \arg \max_{i \in \{1, 2, \dots, k\}} \{p(x \mid c_i)\}$$

# Bayesovská naivní klasifikace

- grafická interpretace typická pro Bayesův naivní klasifikátor vlastně není
- výstupem pro hodnocení přesnosti predikce modelu je konfuzní matice

		přiřazená hodnota			
		1	2	...	$k$
skutečná hodnota	1	$n_{11}$	$n_{12}$	$\cdots$	$n_{1k}$
	2	$n_{21}$	$n_{22}$	$\cdots$	$n_{2k}$
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	$k$	$n_{k1}$	$n_{k2}$	$\cdots$	$n_{kk}$

- přesnost (*accuracy*) vyčíslíme jako podíl stopy a součtu konfuzní matice

$$accuracy = \frac{\sum_{i=1}^k n_{ii}}{\sum_{i=1}^k \sum_{j=1}^k n_{ij}}$$

# Bayesovská naïvní klasifikace v R

- knihovna `e1071`
- funkce `naiveBayes()` s argumenty
  - ▶ `formula` - závislá proměnná a na kterých prediktorech závisí
  - ▶ `data` - dataframe trénovací množiny
- funkce `predict()` s argumenty
  - ▶ `object` - objekty typu model naïvní Bayesovské klasifikace
  - ▶ `newdata` - dataframe testovací množiny
  - ▶ `type` - když `"class"`, jsou vráceny predikované třídy, když `"raw"`; jsou vrácena maxima aposteriorních pravděpodobností
- funkce `table()` pro konfuzní matici

# Bayesovská naivní klasifikace v R

```
## inicializuji balíček "e1071"
suppressWarnings(library("e1071"))

## loaduji data "HouseVotes84"
data(HouseVotes84, package = "mlbench")
head(HouseVotes84[, 1:16], 4)
```

```
##           Class   V1 V2 V3   V4   V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15
## 1 republican     n  y  n     y     y  y  n  n  n  y <NA>   y   y   y   n
## 2 republican     n  y  n     y     y  y  n  n  n  n     n   y   y   y   n
## 3 democrat <NA>   y  y <NA>     y  y  n  n  n  n     y   n   y   y   n
## 4 democrat     n  y  y     n <NA>   y  n  n  n  n     y   n   y   n   n
```

```
## náhodně rozdělují data "HouseVotes84" do trénovací
## a testovací množiny
set.seed(2016)
```

```
train_set_indices <- sample(1:dim(HouseVotes84)[1],
                           floor(0.6 * dim(HouseVotes84)[1]),
                           replace = FALSE)
```

```
train_set <- HouseVotes84[train_set_indices, ]
test_set <- HouseVotes84[-train_set_indices, ]
```



# Bayesovská naivní klasifikace v R

```
## vytvářím model
my_bayes <- naiveBayes(Class ~ ., data = train_set)
```

```
## a dívám se na první z predikovaných hodnot
head(predict(my_bayes, test_set, type = "class"))
```

```
## [1] republican democrat democrat republican republican republican
## Levels: democrat republican
```

```
head(predict(my_bayes, test_set, type = "raw"), 3)
```

```
##          democrat republican
## [1,] 2.506697e-08 0.9999999749
## [2,] 9.997932e-01 0.0002068392
## [3,] 9.955632e-01 0.0044367600
```

```
## vytvářím a dívám se na konfuzní matici
(confusion_matrix <- table(test_set$Class, predict(my_bayes, test_set)))
```

```
##
##          democrat republican
## democrat          89         13
## republican         5         67
```

# Bayesovská naïvní klasifikace v R

```
## počítám přesnost  
sum(diag(confusion_matrix)) / sum(confusion_matrix)
```

```
## [1] 0.8965517
```

# Rozhodovací stromy

- princip: trénovací množina je postupně rozdělována na stále menší podmnožiny tak, aby v každé podmnožině převládaly prvky jedné třídy
- tedy princip „rozděl a panuj“ („divide and conquer“), metoda známa jako *top-down induction of decision tree* (TDIDT)
- v každé iteraci vyberou některý z atributů a určí její hodnotu tak, že trénovací množina je pak hodnotou této proměnné „nejlépe“ rozdělena ve smyslu některé diskriminační metriky
- vzniká tak graf typu strom

# Rozhodovací stromy

- metriky, které jsou pro atributy maximalizovány

- ▶ Giniho index

$$\text{Giniho index}_i = 1 - \sum_{j=1}^k p_{ij}^2$$

- ▶ informační zisk

$$\text{informační zisk}_i = - \sum_{j=1}^k p_{ij} \log_2 p_{ij}$$

- ▶ deviance

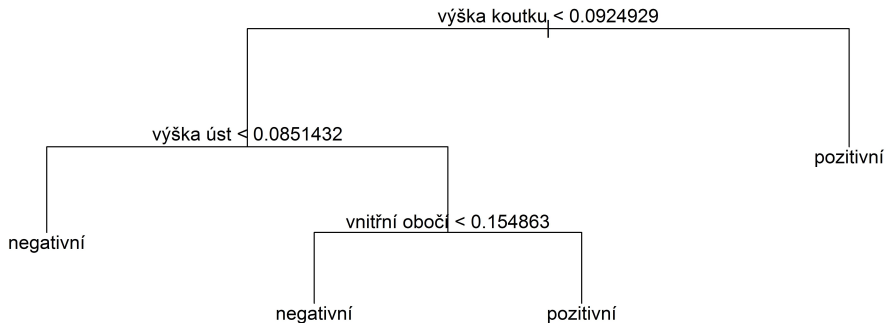
$$\text{deviance}_i = -2 \sum_{j=1}^k n_{ij} \ln p_{ij}$$

- kde  $p_{ij}$  je pravděpodobnost existence  $j$ -té třídy v  $i$ -tém uzlu,  $n_{ij}$  je počet pozorování  $j$ -té třídy v podmnožině  $i$ -tého uzlu,  $k$  je počet tříd

# Rozhodovací stromy

- *pruning* - prořezání výsledného stromu (tj. neuvažování koncových větví stromu od určitého stupně větvení)
  - ▶ pomocí  $k$ -násobné křížové validace
  - ▶ nebo porovnáním nevysvětlené variability vs. počtu uzlů stromu (v diagramu *elbow fenomén*)
  - ▶ apod.

# Rozhodovací stromy



# Rozhodovací stromy v R

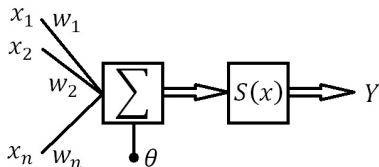
- knihovna `tree`
- funkce `tree()` s argumenty
  - ▶ `formula` - závislá proměnná a na kterých prediktorech závisí
  - ▶ `data` - dataframe trénovací množiny
- funkce `predict()` s argumenty
  - ▶ `object` - objekty typu model naivní Bayesovské klasifikace
  - ▶ `newdata` - dataframe testovací množiny
  - ▶ `type` - když `"class"`, jsou vráceny predikované třídy, když `"raw"`; jsou vrácena maxima aposteriorních pravděpodobností
- funkce `table()` pro konfuzní matici

# Neuronové sítě

- v padesátých letech navržen první model McCullochem a Pittsem

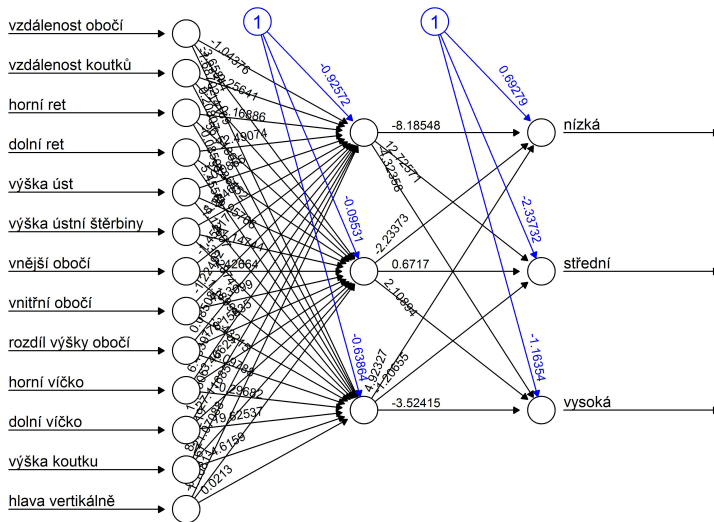
$$Y = S\left(\sum_{i=1}^n (w_i x_i) + \theta\right),$$

kde  $x_i$  jsou vstupy neuronu,  $w_i$  jsou synaptické váhy pro  $i \in \{1, 2, \dots, n\}$ ,  $\theta$  je práh,  $S(x)$  je přenosová, též aktivační funkce neuronu a  $Y$  je výstup neuronu





# Neuronové sítě



# Neuronové sítě v R

- knihovna `neuralnet`
- funkce `neuralnet()` s argumenty
  - ▶ `formula` - závislá proměnná a na kterých prediktorech závisí
  - ▶ `hidden` - počet skrytých vrstev
  - ▶ `linear.output` - zda se jedná o spojitou predikovanou proměnnou
  - ▶ `data` - dataframe trénovací množiny
  - ▶ `threshold` - prah pro prahovou funkci
- funkce `predict()` s argumenty
  - ▶ `object` - objekty typu model naivní Bayesovské klasifikace
  - ▶ `newdata` - dataframe testovací množiny
  - ▶ `type` - když "class", jsou vráceny predikované třídy, když "raw"; jsou vrácena maxima aposteriorních pravděpodobností
- funkce `table()` pro konfuzní matici

# Hands-on! Your turn!

- sámplová data, skrípty a tato prezentace na adrese

<https://github.com/LStepanek/Uvod-do-machine-learning-v-R/>

Děkujeme za pozornost!

`ondrej.klempir@fbmi.cvut.cz`

`lubomir.stepanek@fbmi.cvut.cz`