

# Rozpoznávanie obrazcov - 10. cvičenie

## Rozhodovacie stromy

Viktor Kocur

[viktor.kocur@fmph.uniba.sk](mailto:viktor.kocur@fmph.uniba.sk)

DAI FMFI UK

27.4.2020

# Vyhodnocovanie

## Viacero tried

Doteraz sme väčšinou riešili klasifikačné úlohy, v ktorých sme mali zväčša len dve triedy. Niektoré klasifikátory (NB, kNN), ktoré sme skúšali však dokážu klasifikovať do viaceru tried. Pre tie ktoré sú z princípu len binárne je možné nakombinovať viaceru binárnych klasifikátorov aby sme získali viactriedny klasifikátor.

## fitceocc

$Mdl = \text{fitcecoc}(X, y)$  - vráti multitriednu verziu SVM klasifikátora

# Presnosť

## Stačí presnosť?

Presnosť (accuracy) definujeme ako podiel počtu správne klasifikovaných príkladov s celkovým počtom príkladov. Takáto metrika na klasifikátor však môže zlyhať. Napríklad ak máme v príkladoch iba 10 % jednej triedy a 90 % druhej, tak klasifikátor, ktorý vždy vráti druhú triedu bude mať 90 % presnosť, ale ja nám jasné, že to nieje dobrý klasifikátor.

# Matica zámen

## Matica zámen

Jeden zo spôsobov ako vyhodnocovať klasifikátor je tzv. matica zámen. Matica má v  $i$ -tom riadku a  $j$ -tom stĺpci počet objektov, ktorých sú z triedy  $i$  a sú klasifikované ako trieda  $j$ .

## confusionmat

$C = \text{confusionmat}(g1, g2)$  - vráti maticu zámen pre pravé triedy  $g1$  a predikované triedy  $g2$ .

## confusionchart

$cm = \text{confusionchart}(g1, g2)$  - rovno nakreslí maticu zámen

## True/False Positive/Negative

Budeme teraz operovať s pojmy vždy pre jednu triedu:

- True Positive - TP  
klasifikátor určil, že príklad je z triedy a je to pravda
- False Positive - FP  
klasifikátor určil, že príklad je z triedy ale nieje to pravda
- True Negative - TN  
klasifikátor určil, že príklad nieje z triedy a je to pravda
- False Negative - FN  
klasifikátor určil, že príklad nieje z triedy ale nieje to pravda

# Precision a Recall

## Precision

Presnosť (precision) definujeme ako  $\frac{TP}{TP+FP}$ . Rozdiel medzi precision a accuracy bežne spočíva v tom, že pri accuracy delíme počtom všetkých príkladov.

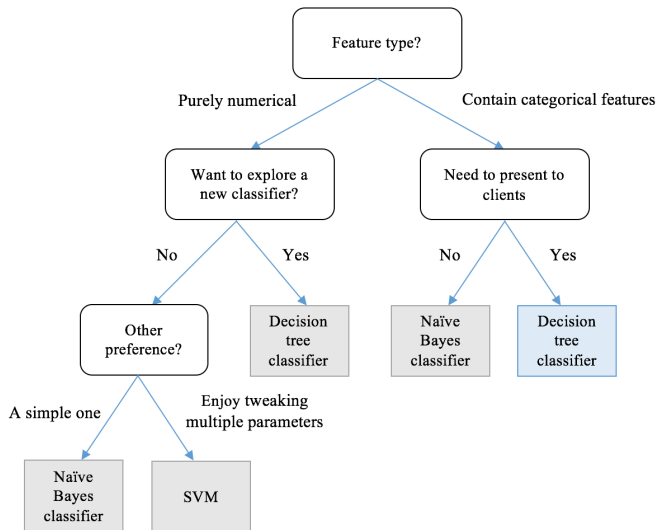
## Recall

Citlivosť (recall) počítame ako  $\frac{TP}{TP+FN}$ , teda podiel koľko príkladov klasifikátor určil správne zo všetkých ktoré mal určiť.

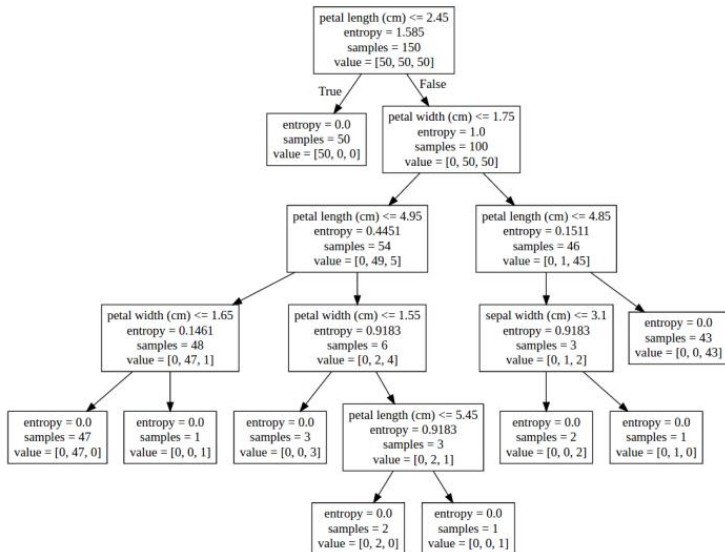
## Úloha

Pre dataset fisheriris vytvorte klasifikátor a spočítajte maticu zámen. Spočítajte aj presnosť a citlivosť pre všetky triedy.

# Rozhodovacie stromy



# Rozhodovacie stromy





# Konštrukcia rozhodovacích stromov

## Rozdelujúce kritérium

Strom konštruujeme, tak že vyberáme príznak a jeho hodnotu na základe ktorého rozdelíme množinu prvkov na dve časti. Tento postup opakujeme s oboma podmnožinami až kým nieje splnené ukončujúce kritérium.

## Ukončujúce kritérium

Môže to byť napríklad: podmnožiny obsahujú iba po jednej triede, strom dosiahol nastavenú hĺbku, menší ako prahový počet zle klasifikovaných prvkov v nejakom uzle, ohodnotenie najlepšieho príznaku je menšie ako prah.

# Rozhodovacie kritériá

## ID3

Vyberáme príznak pre ktorý bude entropia minimálna, teda taký pre ktorý je informačný prínos najväčší (vzájomná informácia s triedami je najväčšia).

## C4.5

Obdobne ako pri ID3, ale tentokrát maximalizujeme normalizovaný informačný prínos. C4.5 navyše dokáže pracovať s numerickými dátami.

# Rozhodovacie kritériá - teória zo 4. cvičenia

## Entrópia

$$H(Y) = \sum_{y \in \omega} -P(Y = y) \cdot \log_2(P(Y = y))$$

## Špecifická podmienená entrópia

$$H(Y|X = v) = H(Y), \text{ len pre hodnoty } Y, \text{ kde } X = x$$

# Rozhodovacie kritériá - teória zo 4. cvičenia

## Vzájomná informácia, informačný prínos

$$I(Y; X) = H(Y) - H(Y|X) = H(Y) - \sum_{x \in \omega} P(X = x) \cdot H(Y|X = x)$$

## Normalizovaný informačný prínos

$$nl(Y; X) = \frac{I(Y; X)}{H(X)}$$

# Príklady

## ID3

[https://sefiks.com/2017/11/20/  
a-step-by-step-id3-decision-tree-example/](https://sefiks.com/2017/11/20/a-step-by-step-id3-decision-tree-example/)

## C4.5

[https://sefiks.com/2018/05/13/  
a-step-by-step-c4-5-decision-tree-example/](https://sefiks.com/2018/05/13/a-step-by-step-c4-5-decision-tree-example/)

# Matlab

## fitctree

$Mdl = \text{fitctree}(X,y)$  - vráti klasifikačný model rozhodovacieho stromu.

## fitctree

$Mdl = \text{fitctree}(T, \text{property})$  - vráti klasifikačný model rozhodovacieho stromu podľa tabuľky  $T$  pre klasifikačný cieľ v stĺpci  $\text{property}$ .

## CART

MATLAB používa metódu CART, ktorá je podobná metóde ID3, ale je mierne iná. Keďže na prednáške nieje, tak ju nebudeme rozoberať.

# Matlab

## predict

Mdl.predict(x) - vráti predpoveď modelu pre daný príznačkový vektor.

## view

Mdl.view('Mode','graph') - zobrazí strom

## Úloha

Vytvorte a zobrazte si strom pre databázu fisheriris a census1994. Pre census1994 zistite presnosť.

# Orezávanie stromov

## Orezávanie

Strom môže byť zbytočne komplikovaný. To vedie na overfitting. Strom je možné orezať tak, že podstromy, ktoré prinášajú zanedbateľné zlepšenie presnosti klasifikácie nahradíme listom.

## prune

$MdIP = \text{prune}(Mdl, 'Property', \text{value})$  - vráti orezaný strom podľa toho ako je nastavená property

## Úloha

Orežte strom pre dáta fisheriris a census1994. Otestujte rôzne properties (Level, Alpha, Nodes) a otestujte zlepšenie presnosti na testovacej množine pre census1994.