Rozpoznávanie obrazcov - 8. cvičenie
Validácia a Rozhodovacie stromy

Viktor Kocur
viktor.kocur@fmph.uniba.sk

DAI FMFI UK

16.4.2019

Rozdelenie dát

Trénovacia množina

Doteraz sme vždy operovali s trénovacou množinou. Teda všetky dáta sme použili na nastavenie parametrov modelu.

Testovacia množina

V prípade, že chceme overiť že náš model je spoľahlivý je nutné odložiť si časť dát na testovanie. Testovacie dáta použijeme až na úplnom konci keď máme model hotový. Používame ich čisto na vyhodnotenie a nie na určenie metódy, alebo parametrov a hyperparametrov modelu.

Rozdelenie dát

Validačná množina

Keďže testovaciu množinu nepoužívame na určenie modelu, tak potrebujeme ešte jednu množinu na tento účel. Validačnú množinu používame na určenie správneho prístupu a nastavenie hyperparametrov modelu.

Rozdelenie dát

Podiely na rozdelovanie dát záležia od ich charakteru, množstva a modelu. Pri neurónových sieťach potrebujeme veľa trénovacích dát, preto je vhodné využiť split 80/10/10. Pri metódach aké sme si zatiaľ ukázali stačí aj 60/20/20. V niektorých prípadoch však je nutné isť ešte ďalej. Existujú datasety kde je split napr. 40/20/40.

Validácia - postup

Hyperparametre

Na validačnej množine určujeme hyperparametre. To sú parametre/nastavenia, ktoré menia spôsob akým sa model trénuje a ako funguje predikcia. Pre SVM je to napr. výber kernelovej funkcie a jej škály. Pre kNN je to napríklad hodnota k a výber metriky.

Validácia

Pre rôzne hyperparametre natrénujeme (v prípade kNN len vytvoríme) na trénovacej množine naše modely. Tieto potom otestujeme na validačnej množine. Použijeme na to nejakú mieru spoľahlivosti. Ideálne presnosť klasifikácie. Na základe výsledkov vyberieme hyperparametre.

Validácia - úloha

Úloha

Rozdelte si dáta z predchádzajúceho cvičenia na train/val/test s pomerom 60/20/20. A určite najlepší parameter k pre kNN klasifikátor a metriku na validačnej množine.

Pozor na dostatočnú reprezentáciu

Často sú dáta zoradené podľa triedy, alebo v nejajek inej pravidelnej forme. Je preto nutné overiť si, či je rozdelenie na train/val/test zmysluplné. Ideálne chceme rovnaký počet tried pre každú množinu.

Vzájomná validácia

Vzájomná validácia

Ak máme málo dát tak nedelíme dáta na trénovacie a validačné. Dáta rozdelíme na n približne rovnakých podmnožín. Model vždy natrénujeme na dátach zo všetkých okrem jednej podmnožiny a otestujeme na jednej podmnožine. Toto opakujeme n krát a výsledok spriemerujeme.

Matlab

```
Mdl = fitcknn(X, y, 'NumNeighbors', k);
CVMdl = crossval(Mdl)
loss = kfoldLoss(CVMdl)
```

Vzájomná validácia

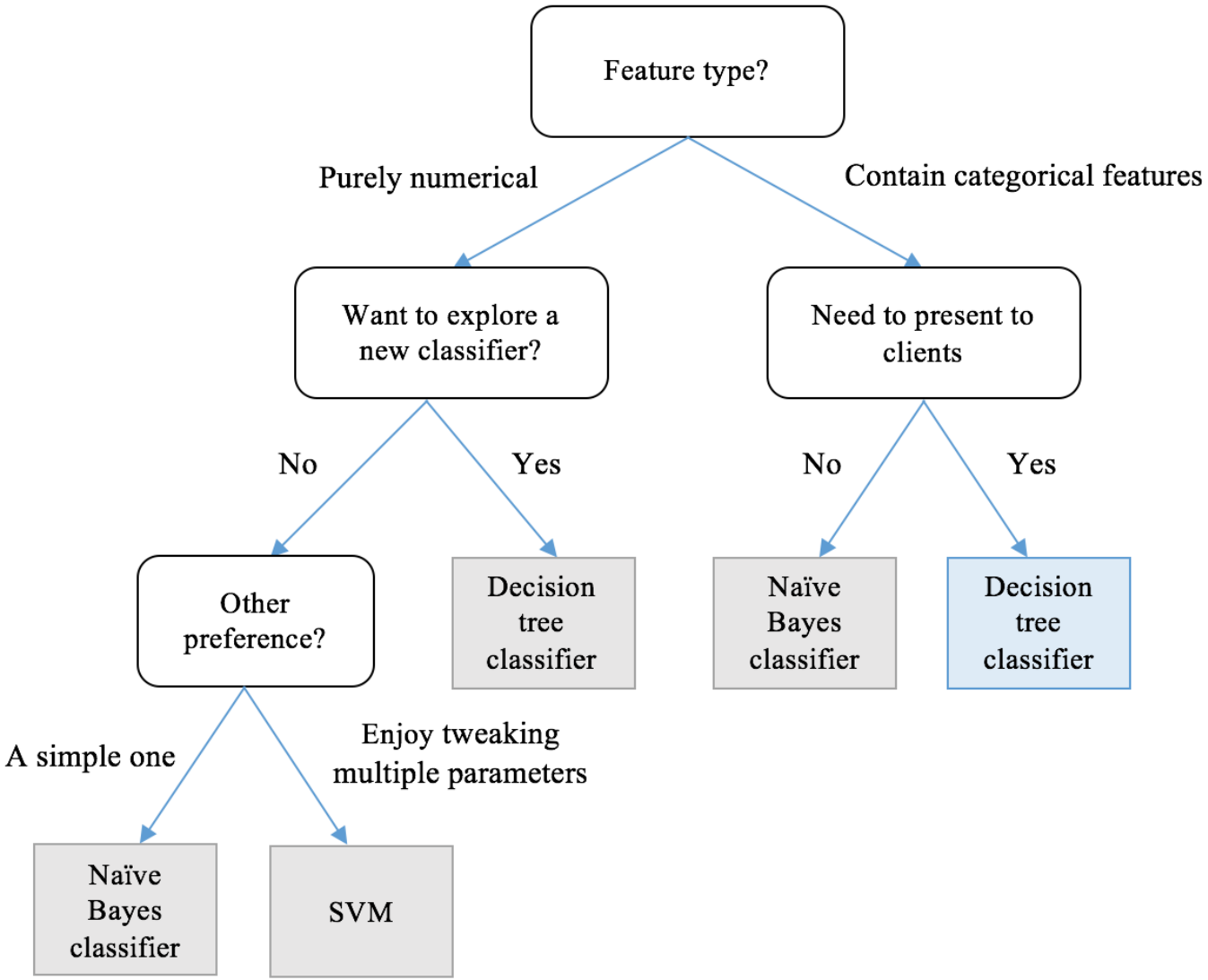
Automatické určenie hyperparametrov

Matlab pri väčšine fitc... funkcií dokáže nájsť optimálne hyperparametre sám. Ak to budete používať je dobre pozrieť sa do helpu.

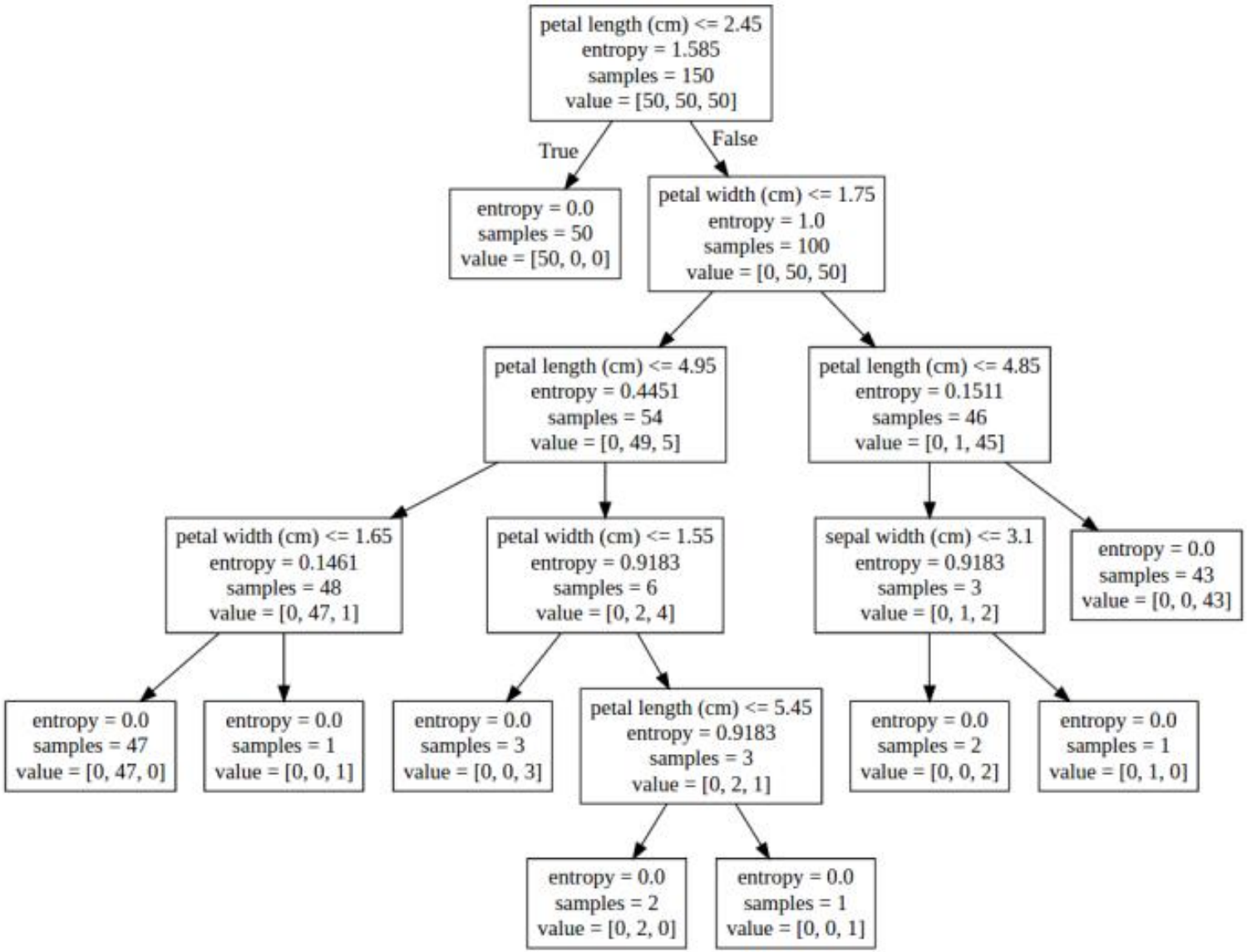
Matlab

```
Mdl = fitcknn(X,Y,'OptimizeHyperparameters','auto')
```

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 entrópia minimálna, teda taký pre ktorý je informačný prínos najväčší (vzájomná informácia s triedami je najväčšia).

C4.5

Obdobne ako pri ID3, ale tentokrát maximaluzujeme 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

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

Príklady

ID3

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/

Matlab

fitctree

Mdl = fitctree(X,y) - vráti klasifikačný model rozhodovacieho stromu.

fitctree

Mdl = fitctree(T,property) - vráti klasifikačný model rozhodovacieho stromu podľa tabulky T pre klasifikačný ciel v stĺpci 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 klasifikačný model rozhodovacieho stromu podľa tabulky T pre klasifikačný ciel v stĺpci property.

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

MdlP = prune(Mdl,'Property', 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.