Rozpoznávanie obrazcov - 8. cvičenie
Validácia a Rozhodovacie stromy

Viktor Kocur
viktor.kocur@fmph.uniba.sk

DAI FMFI UK

9.4.2019

Bayesovo pravidlo

Bayesovo pravidlo

Budeme opäť používať Bayesovo pravidlo:

$$P(\omega_i|\vec{x}) = \frac{P(\vec{x}|\omega_i)P(\omega_i)}{P(\vec{x})} \tag{1}$$

Naivita

Náš klasifikátor je naivný a predpokladá, že príznaky sú nezávislé:

$$P(\vec{x}|\omega_i) = \prod_k P(x_k|\omega_i) \tag{2}$$

Klasifikátor

Klasifikácia

Klasifikujeme pomocou nájdenia triedy s najväčšou pravdebodobnosťou:

$$pred_i = \arg\max_i \left(\frac{P(\vec{x}|\omega_i)P(\omega_i)}{P(\vec{x})} \right) \tag{3}$$

$$= \arg\max_i \left(P(\vec{x}|\omega_i)P(\omega_i) \right) \tag{4}$$

$$= \arg\max_i \left(P(\omega_i) \prod_k P(x_k|\omega_i) \right) \tag{5}$$

Klasifikátor

Výpočet hodnôt

Budeme predpokladať že máme kategorické príznaky. Teda pre každé k môže x_k nadobúdať iba konečne mnoho diskrétnych hodnôt. Označíme celkový počet prvkov trénovacej množiny ako N. Počet prvkov, ktoré patria do triedy ω_i ako N_i. Počet prvkov, ktoré patria do ω_i a pre k-tý príznak majú hodnotu v ako $N_{i,k,v}$. Potom môžeme definovať:

$$P(\omega_i) = \frac{N_i}{N} \tag{6}$$

$$P(x_k = v|\omega_i) = \frac{N_{i,k,v}}{N_i} \tag{7}$$

Klasifikátor

age	income	student	credit rating	buys computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Úloha

Spočítajte do ktorej kategórie bude patriť zákazník s náhodnými prediktormi.

Klasifikátor

Nekategorické dáta

V prípade, že niektorý príznak je numerický, tak nemôžeme aplikovať výpočet z predchádzajúceho slidu. Preto budeme pravdepodobnosť $P(x_k|\omega_i)$ odhadovať nejakou distribučnou funkciou.

Parametrické metódy

Pri parametrických metódach odhadneme parametre nejakého dopredu určeného rozdelenia.

Neparametrické metódy

Pri neparametrických metódach pravdepodobnosť vypočítame na základe bodov z trénovacej množiny v okolí bodu o ktorý sa zaujímame.

Matlab

fitcnb

Mdl = fitcnb(T,'nazov_pola') - vráti naivný Bayesov klasifikátor pre tabuľku T pre klasifikačný cieľ pre stĺpec nazov_pola.

Na dátach

```
load census1994
Mdl = fitcnb(adulddata, 'salary');
```

Úloha

Zistite presnosť klasifikátora tak, že ho spustíte (Mdl.predict) na tabuľku adulttest a porovnáte výsledok.

Matlab

fitcnb

Mdl = fitcnb(X,y) - vráti naivný Bayesov klasifikátor

Úloha

Otestujte naivný Bayesov klasifikátor na fisheriris dátach.

Úloha

Zobrazte si klasifikátor na dátach zo 6. cvičenia pomocou úpravy skriptu showSVM z toho istého cvičenia.

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')
```