Rozpoznávanie obrazcov - 8. cvičenie
Validácia a One-hot kódovanie

Viktor Kocur
viktor.kocur@fmph.uniba.sk

DAI FMFI UK

13.4.2020

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')
```

Kategorické dáta

Kategorické dáta

Niekedy dostaneme dáta v tzv. kategorickej forme. Teda jeden z príznakov môže byť len z nejakej konečnej množiny možností. Napr. áno/nie, študent/pracujúci/dôchodca/nezamestnaný atď. Niektoré metódy ktoré sme si zatiaľ ukázali nevedia s takýmito dátami pracovať. Konkrétne ide o Lineárny klasifikátor/SVM. kNN v Matlabe dokáže operovať s kategorickými premennými pomocou špeciálnej metriky, ale je nutné to nastaviť.

Problém s jednoduchou konverziou

Prečo nepoužiť numerické dáta vždy

Jeden jednoduchý spôsob konverzie z kategorických dát na numerické by bolo, že prekonvertujeme dáta na numerické, tak že postupne priradíme každej kategórii číslo. To však nieje vhodný postup. Jeden z dôvodov je, že napríklad ak napríklad máme kategórie chodec, cyklista, motorka, auto, dodávka, kamión. A priradíme im čísla chodec:0, cyklista:1, ... kamíon:5. Tak by nám vyšlo, že priemer kamióna a cyklistu je auto, čo nedáva zmysel. Takýto postup by však mal zmysel na príklad v situácii ak máme katogórie ako známky A/B/C/D/E/Fx. U nich totiž približne platí, že C je priemer B a D atď.

One-hot kódovanie

One-hot kódovanie

Aby sme sa vyhli problému z predchádzajúceho slidu, tak použijeme tzv. one-hot kódovanie. Každý kategorický príznak, ktorý ma m možných hodnôt transformujeme na príznakový vektor dĺžky m, tak že každý prvok vektoru bude korešpondovať jednej kategórii a nastavíme ho ako 1 ak v pôvodnom zápise ma príklad danú kategóriu. Ostatné prvky nastavíme na nula. Takže napríklad ak máme dáta v tvare rýchlosť a typ auta: (20, cyklista), (58, auto), tka nám vzniknú vektory: (20, 0, 1, 0, 0, 0, 0) a (58, 0, 0, 0, 1, 0, 0).

One-hot kódovanie Matlab

dummyvar

d = dummyvar(c) - vráti one-hot kódovanie pre stĺpec kategorických príznakov c

categorical

c = categorical(r) - categorical vráti keategorický dátový typ pre vektor r. Používame najmä na konverziu z cell štruktúry do categorical.

categories

cats = categories(c) - vráti kategórie v kategorickom vektore c.

One-hot kódovanie Matlab

load patients

Načítajte si dáta kardiologigkých pacientov pomocou load patients.

Úloha

Prekonvertujte kategorické príznaky na one-hot kódovanie a natrénujte na datasete SVM. Informáciu o mene a nemocnici nepoužívajte. Cieľom predikcie bude určiť, či je pacient fajčiar.