2. domača naloga: Informativnost značilk pri napovedovanju posameznih tem

Miha Zidar (63060317)

5. marec 2012

1 Uvod

V tej nalogi smo si pogledali odvisnost razredov in atributov. Začeli smo z diskretizacijo vseh vrednosti na 2 dela; elementi večji od 0 in elementi enaki 0. Nato smo se za vsak par atributov z pomočjo permutacijskega testa odločili ali obstaja dovolj zanesljiva povezava med vrednostjo atributa in vrednostjo razreda.

2 Metode

2.1 Infomacijski prispevek

Za ocenjevanje kakovosti atributov smo uporabili mero infomacijski prispevek (information gain) med atributom in razredom.

Definicija

Enačba za izračun informacijskega prispevka je:

$$\mathrm{Gain}(Y,X) \;=\; \mathrm{H}(Y) - \mathrm{H}(Y|X) \;=\; \mathrm{I}(Y,X)$$

$$\mathrm{I}(Y,X) \;=\; \sum_{y\in Y}\sum_{x\in X} p(x,y)\log\left(\frac{p(x,y)}{p(x)*p(y)}\right)$$

Primer

Podani imamo dve naključni spremenljivki, ki v nasi nalogi predstavljata dva stolplca v tabeli. X predstavlja naš stolpec pod atributom, Y pa vrednosti v nasem razredu.

$$\begin{aligned} X &= (0,0,1,0,0,1) \\ Y &= (0,1,1,0,0,1) \end{aligned}$$

Sedaj izracunamo vse potrebne verjetnosti.

$$\begin{aligned} p(X{=}0) &= 4/6 \\ p(X{=}1) &= 2/6 \\ p(Y{=}0) &= 3/6 \\ p(Y{=}1) &= 3/6 \\ p(X{=}0,Y{=}0) &= 3/6 \\ p(X{=}0,Y{=}1) &= 1/6 \\ p(X{=}1,Y{=}0) &= 0/6 \\ p(X{=}1,Y{=}1) &= 2/6 \end{aligned}$$

Vstavimo te verjetnosti v zgornjo enacbo in dobimo:

$$3/6 \log\left(\frac{3/6}{(4/6)*(3/6)}\right) + 1/6 \log\left(\frac{1/6}{(4/6)*(3/6)}\right) + 0 + 2/6 \log\left(\frac{2/6}{(2/6)*(3/6)}\right)$$

in octave nam izpise rezultat (kjer aX predstavlja posamezni člen enačbe):

```
octave:19> a1+a2+a3+a4
ans =  0.459147917027245
```

moj program pa za stevili 9 ($0b1001$) in 25 ($0b11001$) vrne 0.459147917027, kar je enako vendar z nekaj decimalnimi mesti manj. Za razliko od Orange orodja ki pa se v povprečju razlikuje že na sedmem decimalnem mestu, kar sem turi preveril z dodatnim testnim programom ki je racunal povprecno odstopanje od mojih izračunanih vrednosti.

2.2 Permutacijski test

Prmutacijski test en izmed načinov kako ugotovimo ali je informacijski prispevek našega atributa za dani razred dober, ali pa je tak le zaradi naključja. To pa ugotovimo enostavno tako, da primerjamo informacijski prispevek našega razreda z mnogimi naključnimi permutacijami tega razreda. Na koncu se odločimo o pomembnosti atributa glede na to, koliko procentov naključnih permutacij je pokazalo višji informacijski prispevek. Tej stopnji zaupanja bomo rekli *alpha* in spodaj imamo prikazane stevilo pomembnih atributov, ki jih dobimo pri različnih vrednosti *alpha*.

Ker smo računali z dosti podatki, smo prišli do časovnega problema, in sicer navaden permutacijski test za 100 permutacij je trajal priblizno 2h. Iz optimizacijskih razlogov sem se zato odločil da ne uporabljam orange, vendar hranim podatke na drugačen način in sam računam informacijski prispevek za te podatke. Tako sem čas izvajanja zmanjšal iz 2h na priblizno 3 minute.

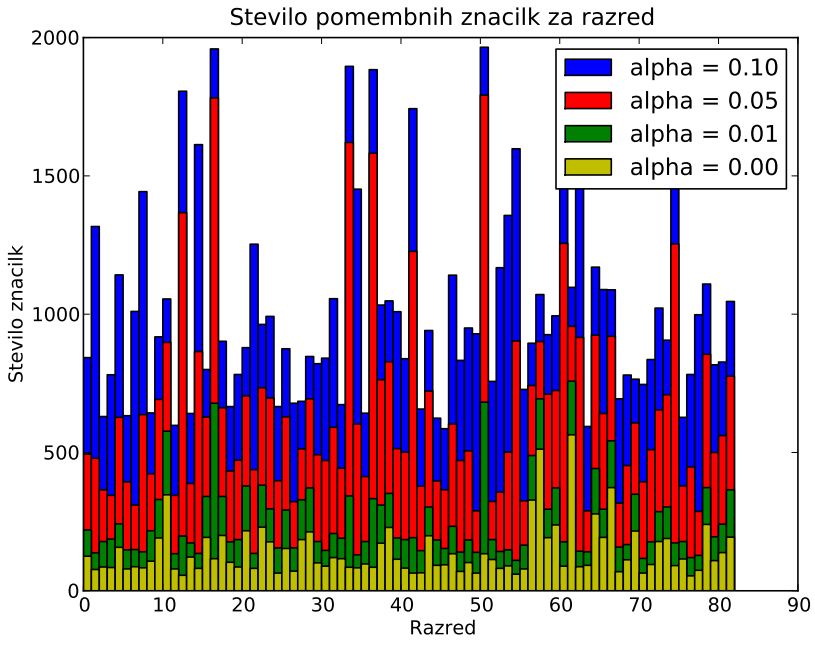
2.3 Program in optimizacija

Ker imamo opravka le z binarnimi atributi, nima smisla vsakega elementa hraniti v seznamu, vendar je prostorsko in procesorsko dosti bolj učinkovito če posamezne stolpce predstavimo kot navadna števila. V taki predstavitvi pa imamo še eno prednost: za štetje kje so enake ali različne vrednosti lahko uporabljamo enostavne bitne operacije, ki so mnogo hitrejše kot operacije nad seznami.

Ko pa imamo podatke lepo predstavljene, pa potrebujemo še nekaj extra podatkov, ker se začetne nule pri številih drugače izgubijo. Tako sem za vsako število hranil koliko je dolgo, koliko enic vsebuje in stevilo samo. Napisal pa sem tudi vse potrebne funkcije za delovanje, Kar si lahko pogledate v kodi.

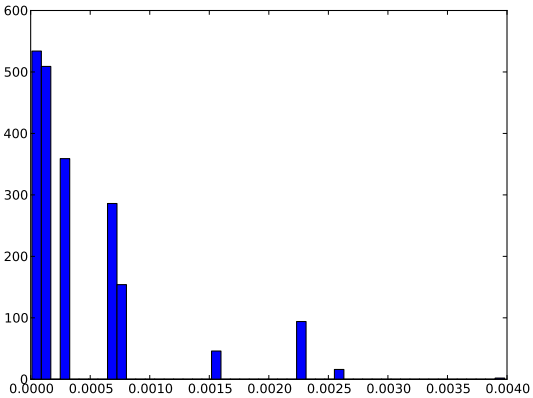
3 Rezultati

Za prikaz rezultatov smo pognali permutacijski test z 500 permutacijami, in rešitev predstavili z histogramom ki prikazuje koliko pomembnih atributov imamo pri razlicnih alpha vrednostih. Histogram bi sicer dosti lepse zgledal ce bi bil urejen po entropiji posameznega razreda, vendar mi tega še ni uspelo dokončati, saj so podatki ločeni med seboj, kar mi je povzročalo težave. V rezultatih pa vidimo kako hitro raste stevilo pomembnih atributov v odvisnosti od alpha, saj ko je alpha 0.1 so pri nekaterih razredih pomembni skoraj vsi atributi.

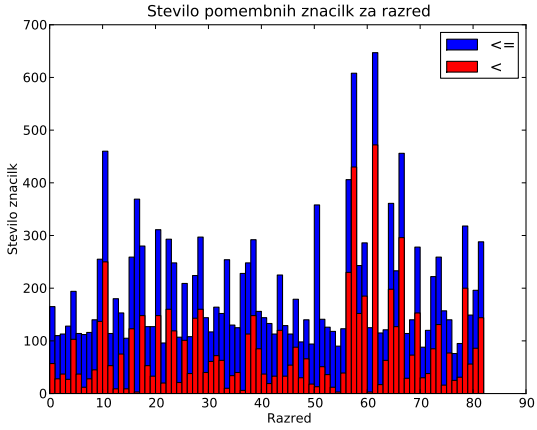


Slika 1: Prikaz stevila pomembnih atributov za posamezne razrede, pri razlicnih alpha vrednostih.

Ena stvar, ki sem jo opazil je, da lahko pri zelo majhni spremembi v kodi (naprimer pogoj za gledanje ali je razred pomemben zamenjamo iz <= na <) lahko dobimo zelo velike spremembe v rezultatih (v enem primeru je stevilo znacilk padlo iz 300 na 50). Zato sem izrisal še histogram ki prikazuje porazdelitev posameznih vrednosti izračunanega informacijskega prispevka pri 2000 permutacijah. Rezultat je bil zelo nepricakovan, vendar je lepo pojasnil velika odstopanja v majhni spremembi pogoja.



Slika 2: porazdelitev info gain za razred c40 pri atributu D_1404, za 2000 permutacij.



Slika 3: Odstopanje stevila značilk za posamezni razredi pri pogojih <= na < za izločanje, ter pri $alpha = 0$.

Izkaže se, da je velikokrat dosti enakih vrednosti, kar vidimo že iz nizke entropije atributov. Iz tega sklepam da bi bilo bolj smiselno te rezultate primerjati še z kakšno drugo mero za ocenjevanje atributov enkrat v prihodnosti, zaenkrat pa so te rezultati čisto zadovoljivi.

4 Izjava o izdelavi domače naloge

Domačo nalogo in pripadajoče programe sem izdelal sam.