Razvrščanje člankov v tematske skupine

Gregor Majcen (63070199)

18. marec 2012

1 Izjava o izdelavi domače naloge

Domačo nalogo in pripadajoče programe sem izdelal sam.

2 Uvod

Tretja domača naloga je namenjena izdelavi sistema za napoved oznak dokumentov tekmovanja RS12Contest. Za lažji začetek je naloga zastavljena na podmnožici vseh podatkov (2000 primerov in 10000 atributov). Narejeno je tudi interno tekmovanje, kjer tekmujemo s sošolci. Vse skupaj je kot predpriprava na resno tekmovanje.

3 Metode

3.1 Ocenjevanje točnosti

3.1.1 Prečno preverjanje

Prečno preverjanje je preverjanje s pomočjo razdelitve učne množice na različne dele. Sam sem implementiral k-fold, kar pomeni, da (kadar je k=10) učno množico razdelimo na 10 približno enakih delov. Ena desetina je namenjena t.i. testnim podatkom, ostalo pa množici, na kateri zgradimo model in ta model testiramo na prejšnji desetini. Ta postopek naredimo 10-krat, tako da je vsak primer natanko enkrat uporabljen za testiranje.

3.1.2 F ocena

Ker nam rezultati prečnega preverjanja ne povedo veliko, jih je pametno oceniti. Ena izmed ocen preverjanja točnosti klasifikatorja je F ocena, katero sem tudi implementiral. Odvisna je od množice T in P. T je množica vseh pravilnih napovedi oznak, P pa množica vseh napovedanih napovedi oznak. Oceno se da zelo enostavno izračunati po spodnji enačbi, kot končna ocena pa je povrečje vseh teh.

$F = 2 * \frac{tocnost * priklic}{tocnost + priklic}$
$tocnost = \frac{T \cap P}{P}$
$priklic = \frac{T \cap P}{T}$

3.2 Napovedni modeli

Za učno množico sem iz prvotnih podatkov odstranil vse atribute, ki imajo manj kot 10 neničelnih vrednosti. S tem sem se znebil približno 80% domnevno neuporabnih atributov. Po zgrajenih modelih je bil največji problem izbrati primerno število razredov in postaviti mejo. Ta problem sem poimenoval CC.

Naivni bayes Naivni bayes je prva metoda, ki sem jo implementiral. S pomočjo prečnega preverjanja sem ugotovil, da so rezultati boljši od večinskega razreda, vendar ga zaradi zelo nenatančnih verjetnosti (zelo blizu 1 ali 0) vseeno nisem uporabil, saj nisem vedel kako naj rešim CC.

Orange multilabel klasifikator CC problem sem prepustil klasifikatorju. Izmed vseh možnih modelov sem izbral tistega, ki se je najbolje obnesel pri 10-kratnem prečnem preverjanju. Rezultati niso bili najboljši, vendar vseeno dovolj dobri, da sem presegel prvi prag.

Random Forest Uporabil sem algoritem, ki je napisan v knjižnjici Orange. Generiral sem 200 dreves, nato sem si pa rezultate zaradi dolgotrajnega izvajanja (približno minuto na drevo) shranil za nadaljno obdelavo. To je tudi razlog zakaj od tu naprej nisem več preverjal rezultatov s prečnim preverjanjem. Reševanje CC problema je na veliko načinov opisan v datoteki main.py (funkcije convert...)

Ostalo Vsi ostali algoritmi (lastna različica KNN, Orange KNN, bayes na različne načine) so implementirani v datoteki main.py, vendar niso prinesli želenih rezulatov.

4 Rezultati

Vsi moji rezultati so na strežniku kaggle pod imenom Gregor Majcen. Najboljši rezultati so iz random forest modela. Kot je že napisano f-ocene tu nisem več računal zaradi predolgotrajnega dela. Številke označene z * so bile oddane pred 12.3.

Tabela 1: Rezultati RandomForest (200 dreves)

št.	ocena	komentar
1*	0.40815	Izbrani so najboljši štirje razredi za vsak primer.
2*	0.40453	Izbranih je najboljših šest razredov za vsak primer. Vsak razred je moral imeti verjetnost večjo od praga 0.2
3*	0.41016	Izbrani so najboljši štirje razredi za vsak primer. Vsak razred je moral imeti verjetnost večjo od praga mediane vseh četrtih razredov, če to ni bilo izpolnjeno vzami najboljše tri
4*	0.43446	Izbrani so vsi razredi, ki so bili za 30% slabši od najboljšega. Preveliko število napovedi sem rešil s kaznovanjem vsakega naslednika. Funkcija je napisana main.py: finalConvert()
5	0.41379	Lasten stacking vseh prej opisanih modelov