SVEUČILIŠTE U ZAGREBU

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

Metode prepoznavanja tematike teksta na osnovu sadržaja

Ana Beblek
Barbara Carević

Voditelj: Mr. sc. Mile Šikić

2008.

Sadržaj

1 Uvod 3

2 Klasifikacija teksta 4

2.1 Definicija klasifikacije teksta 4

2.2 Single label i multilabel klasifikacija teksta 4

2.3 Hard i soft klasifikacija teksta 5

2.4 Dokumentno orijentirana klasifikacija teksta (DPC) i kategorijski orijentirana klasifikacija teksta (CPC) 5

3 Metode odabira svojstava 6

3.1 Učestalost dokumenta 6

3.2 Informacijska dobit 6

3.3 Mjera uzajamne informacije 7

3.4 χ^2 statistika 8

3.5 Povezanost riječi 8

4 Metoda k-najbližih susjeda 10

4.1 Algoritam k-najbližih susjeda 11

4.2 Modifikacija algoritma k–najbližih susjeda uvođenjem težinskih faktora 12

4.3 Karakteristike algoritma k – najbližih susjeda 12

5 Stabla odluke 13

5.1 Problemi prikladni za rješavanje pomoću stabla odluke 13

5.2 Predstavljanje stabla odluke 13

5.3 Osnovni algoritam učenja stabla odluke 15

5.4 Karakteristike algoritma ID3 17

6 Naivni Bayesov klasifikator 18

6.1 Bayesov teorem 18

6.2 Naivni Bayesov klasifikator – teorija 19

6.3 Primjena Naivnog Bayesovog klasifikatora u klasifikaciji teksta 20

6.4 Karakteristike Naivnog Bayesovog klasifikatora 21

7 Metoda potpornih vektora - SVM 22

7.1 Klasifikator s maksimalnom marginom 23

7.2 Metoda potpornih vektora sa slabom marginom 26

7.3 Karakteristike metodu potpornih vektora (SVM) 28

8 Umjetne neuronske mreže 29

8.1 Model umjetne neuronske mreže 29

8.2 Topologija neuronskih mreža 31

8.3 Učenje 31

9 Testiranje 33

10 Literatura 36

Kazalo slika

Slika 1. Primjer klasifikacije knn algoritmom za k=3 i k=5 11
Slika 2. Dijagram stabla odluke 14
Slika 3. Primjer stabla odluke 14
Slika 4. Rad stabla odluke.. 15
Slika 5. Primjer entropije 16
Slika 6. Preslikavanje pokaznih uzoraka iz ulaznog prostora (eng. *input space*) u *N*-dimenzionalni prostor *F* (eng. *feature space*) 22
Slika 7. Klasifikator s maksimalnom marginom.. 23
Slika 8. Slojevi umjetne neuronske mreže 29
Slika 9. Topologija neuronskih mreža 31

Kazalo tablica

Tablica 1. Decizijska matrica.. 4
Tablica 2. Usporedni rezultati različitih klasifikatora ispitanih na pet različitih verzija baza novinskih članaka Reuters. Oznaka "F1" označava uporabu F1 mjere učinkovitosti (van Rijsbergen, 1972, 1979 [8]; Lewis, 1995 [9]), oznaka "M" označava makro usrednjavanje (eng. *macroaverage*)... 34

1 Uvod

Klasifikacija teksta prema sadržaju je problem koji je postao aktualan u posljednje vrijeme zbog velike količine dostupnih tekstova u elektroničkom obliku. Smatra se da je danas preko 80% informacija pohranjeno u tekstualnom obliku., zbog te velike i brzo rastuće količine podataka isključeno je korištenje ljudi na tom poslu pa se krenulo prema metodama za automatsku klasifikaciju teksta. Klasifikacija teksta je jedan od problema dubinske analize teksta (eng. *text mining*), koja je interdisciplinarno polje istraživanja dubinske analize podataka (eng. *data mining*), dohvata podataka (eng. *informational retrieval*), strojnog učenja (eng. *machine learning*), statistike i računalne lingvistike (eng. *computational linguistics*).

U ovom seminaru je dan pregled najvažnijih metoda za automatsku klasifikaciju teksta u predefinirane kategorije.

2 Klasifikacija teksta

2.1 Definicija klasifikacije teksta

Neka je $C = \{c_1, c_2, ... c_m\}$ skup predefiniranih kategorija (klasa) i $D = \{d_1, d_2, ... d_n\}$ skup dokumenata. Klasifikacija teksta je proces dodjeljivanja vrijednosti svakom paru $\{d_j, c_i\} \in D \times C$. Ta vrijednost može biti tipa Boolean, nula ako dokument ne pripada u tu kategoriju i jedan ako pripada ili u rasponu od 0 do 1, gdje vrijednost određuje prikladnost neke određene kategorije.

Pridruživanje se može predstaviti decizijskom matricom.

Tablica 1. Decizijska matrica

	d_1	**...**	**d_j**	**...**	**d_n**
c_1	a_{11}	...	a_{1j}	...	a_{1n}
...	...	...	...	...	...
c_i	a_{i1}	...	a_{ij}	...	a_{in}
...	...	...	...	...	...
c_m	a_{m1}	...	a_{mj}	...	a_{mn}

Formalno klasifikacija teksta je proces aproksimacije nepoznate ciljne funkcije, koja opisuje kako bi dokumenti trebali biti klasificirani, pomoću funkcije $\Phi : D \times C \rightarrow \{T, F\}$ (klasifikator, pravilo, hipoteza, model)

Nazivi kategorija nisu bitni, to su samo simboličke labele koje ne utječu na klasifikaciju, sva klasifikacija se vrši samo iz podataka dobivenih iz samog dokumenta.

2.2 Single label i multilabel klasifikacija teksta

Kategorije se mogu i ne moraju preklapati, tj. jedan dokument može pripadati u više kategorija.

Razlikujemo dva slučaja. Single-label TC (eng. *single-label text categorization*) je slučaj kad svakom dokumentu može biti dodijeljena samo jedna kategorija, kategorije se ne preklapaju. Multilabel TC (eng. *multilabel text categorization*) je slučaj kad bilo koji broj kategorija može biti dodijeljen istom dokumentu, svaki dokument može imati nula, jednu ili više kategorija. Kategorije se preklapaju. Metode klasifikacije za single-label TC u multilabel TC se ponešto razlikuju, pa je jako bitno dovoljno rano odrediti koja je prikladnija za određeni problem.

2.3 Hard i soft klasifikacija teksta

Ukoliko se donosi binarna odluka pripada li dokument određenoj kategoriji, to se naziva čvrsta (eng. *hard*) kategorizacija teksta. Mekana (eng. *soft*) kategorizacija ocjenjuje prikladnost neke kategorije dokumente, pridružujući mu brojčanu vrijednost između 0 i 1.

2.4 Dokumentno orijentirana klasifikacija teksta (DPC) i kategorijski orijentirana klasifikacija teksta (CPC)

Kod dokumentno orijentirane klasifikacije (eng. *category –pivoted categorization*) za odabrani dokument $d_j \in D$, pronalaze se sve kategorije $c_i \in C$ gdje bi se on trebao svrstati, decizijska matrica se popunjava po redcima. Kod kategorijski orijentirane klasifikacije (eng. *document–pivoted categorization*) za odabranu $c_i \in C$ kategoriju se pronalaze svi dokumenti $d_j \in D$ koji joj pripadaju, decizijska matrica se popunjava po stupcima. Razlika je bitna jer skupovi D i C nisu uvijek od početka dostupni. DPC je pogodniji kada dokumenti postaju dostupni jedan po jedan (e-mail), dok je CPC prikladniji pri dodavanju novih kategorija c u skup C postojećih kategorija, nakon što je dio dokumenata već klasificiran (klasificiranje web stranica). U praksi je CPC većinom bolja opcija.

3 Metode odabira svojstava

Metode odabira svojstava za pojedine algoritme za klasifikaciju teksta igraju važnu ulogu pri klasifikaciji. Odabirom svojstava povećava se učinkovitost algoritama. Mnogi algoritmi imaju poteškoće pri klasifikaciji teksta s velikim prostorom svojstava (eng. *feature* space) koji se može smanjiti primjenom metoda. U nastavku slijedi opis najvažnijih metoda odabira svojstava.

3.1 Učestalost dokumenta

Učestalost dokumenta (eng. *document frequency* - *DF*) je broj dokumenata u kojima se pojavljuje određeni izraz (riječ ili fraza). Prilikom kategorizacije teksta, izbacuju se oni izrazi čiji je DF je manji od definiranog praga. Pretpostavlja se da takvi izrazi ne utječu na kategorizaciji, a njihovim uklanjanjem se pridonosi smanjenju prostora svojstava, time i povećanju točnosti kategorizacije. Ova metoda odabira svojstava je najjednostavnija, ali predstavlja ad hoc pristup povećanju efikasnosti kategorizacije.

3.2 Informacijska dobit

Informacijska dobit (eng. *information gain* - *IG*) predstavlja rezultat statističkog testa koji se računa na osnovu prisutnosti određenog izraza u dokumentu. Često označava kriterij za dobrotu (eng. *goodness*) izraza u području strojnog učenja.

Neka $\{c_i\}_{i=1}^m$ označava grupu kategorija u ciljnom prostoru. Informacijska dobit G izraza t jednaka je:

$$\begin{aligned} G(t) = &-\sum\nolimits_{i=1}^m P_r(c_i)\log P_r(c_i) \\ &+ P_r(t)\sum\nolimits_{i=1}^m P_r(c_i\,|\,t)\log P_r(c_i\,|\,t) \\ &+ P_r(t)\sum\nolimits_{i=1}^m P_r(c_i\,|\,t)\log P_r(c_i\,|\,t) \end{aligned} \qquad [1]$$

Informacijska dobit se izračunava za svaki pojedini izraz te se iz prostora svojstava se izbacuju oni izrazi koji imaju čija je informacijska dobit manja od predefinirane vrijednosti. Prilikom izračuna uzima se u obzir procjena uvjetnih vjerojatnosti kategorije određenog izraza, te vrijednost entropije. Vremensku složenost procjene vjerojatnosti iznosi $\mathrm{O}(N)$, a prostorna složenost $\mathrm{O}(VN)$, gdje N označava broj dokumenata u skupu za učenje, a V veličinu rječnika izraza. Vremensku složenost proračuna entropije je $\mathrm{O}(Vm)$

3.3 Mjera uzajamne informacije

Mjera uzajamne informacije (eng. *mutual information - MI*) je metoda koja mjeri međusobnu ovisnost dviju varijabli.

Da bi objasnili metodu potrebno je definirati:

- t – izraz
- c – kategorija
- A - broj pojavljivanja izraza t u kategoriji c
- B - broj pojavljivanja izraza t u drugim kategorijama
- C – broj kategorija u kojima se ne pojavljuje izraz t
- N – ukupan broj dokumenata

Definicija mjera uzajamne informacije između t i c glasi:

$$I(t,c) = \log \frac{P_r(t \wedge c)}{P_r(t) \times P_r(c)} \qquad [2]$$

a njena vrijednost se procjenjuje pomoću:

$$I(t,c) \approx \log \frac{A \times N}{(A+C) \times (A+B)} \qquad [3]$$

Ako su t i c nezavisni, vrijednost $I(t,c)$ je nula.

Dobrota izraza pri odabiru svojstava se računa tako da se uzme u obzir izračun mjere za pojedine kategorije na slijedeći način:

$$I_{avg}(t) = \sum_{i=1}^{m} P_r(c_i) I(t,c_i) \qquad [4]$$

$$I_{\max}(t) = \max_{i=1}^{m} \{I(t,c_i)\} \qquad [5]$$

Vremenska složenost mjera uzajamne informacije je $O(Vm)$.

Nedostatak ove metode predstavlja veliki utjecaj rubne vjerojatnosti izraza na izračun mjere, kao što se može vidjeti iz slijedećeg izraza:

$$I(t,c) = \log P_r(t \mid c) - \log P_r(t) \qquad [6]$$

Ako je uvjetnom vjerojatnost $P_r(t \mid c)$ među izrazima jednaka, izrazi će rijetko imati veću mjeru od uobičajenih izraza. Stoga, ne možemo uspoređivati vrijednost mjere među izrazima koji imaju vrlo veliku razliku u učestalosti u kategorijama.

3.4 χ^2 statistika

χ^2 statistika (eng. *CHI*) mjeri nedostatak neovisnosti između izraza *t* i kategorije *c*.

Da bi objasnili metodu potrebno je definirati:

- *t* – izraz
- *c* – kategorija
- *A* - broj pojavljivanja izraza *t* u kategoriji *c*
- *B* - broj pojavljivanja izraza *t* u drugim kategorijama
- *C* – broj kategorija u kojima se ne pojavljuje izraz *t*
- *D* - broj izraza u kategoriji kada se ne pojavljuju niti *t* niti *c*
- *N* – ukupan broj dokumenata

Dobrota izraza je definirana izrazom:

$$\chi^2(t,c) = \frac{N \times (AD - CB)^2}{(A+C)\times(B+D)\times(A+B)\times(C+D)} \qquad [7]$$

Ako su *t* i *c* nezavisni, vrijednost χ^2 statistike je nula.

Računanjem χ^2 statistike između svakog pojedinog izraza i određene kategorije dolazimo do izraza za računanje χ^2 statistike po kategorijama:

$$\chi^2_{avg}(t) = \sum_{i=1}^{m} P_r(c_i)\chi^2(t,c_i) \qquad [8]$$

$$\chi^2_{\max}(t) = \max_{i=1}^{m}\left\{\chi^2(t,c_i)\right\} \qquad [9]$$

χ^2 statistika ima kvadratnu prostornu složenost.

Velika razlika između χ^2 statistike i mjera uzajamne informacije je što se između različitih izraza *t* u istoj kategoriji *c*, χ^2 vrijednosti mogu uspoređivati. Ali, to svojstvo ne vrijedi kod izraza koji imaju malu učestalost u kategorijama.

3.5 Povezanost riječi

Povezanost riječi (eng. *term strength - TS*) predstavlja vjerojatnost da će se izraz pojaviti u usko povezanim dokumentima. Dokumenti u skupu za učenje izdvajaju se u parove ako je njihova sličnost iznad određene vrijednosti. Povezanost riječi se računa pomoću procjene uvjetne vjerojatnosti da se izraz pojavljuje u drugom dijelu para povezanih dokumenata ako se pojavljuje u prvom.

Neka su x i y proizvoljan par različitih, ali povezanih dokumenata, a neka je t izraz. Onda je definicija snage izraza:

$$s(t) = P_r(t \in y \mid t \in x) \qquad [10]$$

Ova mjera odabira svojstava je potpuno različita od gore navedenih mjera. Zasniva se na grupiranju dokumenata uz pretpostavku da su dokumenti koji sadrže mnoge slične riječi i sami slični, a te riječi sadrže velik dio informacije. Ova metoda ne uzima u obzir povezanost između izraza i kategorije. Po tome je slična metodi učestalosti dokumenata, a znatno se razlikuje od, na primjer, informacijske dobiti ili mjere uzajamne informacije.

4 Metoda k-najbližih susjeda

Metoda k–najbližih susjeda potpada pod metode učenja na temelju primjera. Primjeri su pohranjeni, a postupak generalizacije je odgođen do trenutka potrebe za klasifikacijom novog uzorka. Tada se određuje vrijednost ciljne funkcije ispitivanjem odnosa novog uzorka prema pohranjenom primjeru. Takve metode kod kojih se odluka o klasifikaciji odgađa do predočavanja novog primjera nazivaju se lijene metode (eng. *lazy methods*).

U algoritmu k–najbližih susjeda primjeri su točke u n-dimenzionalnom prostoru R_n. Udaljenost između primjera se računa Euklidskom metrikom, a ciljna funkcija može imati diskretne ili realne vrijednosti.

Primjer *x* je opisan vektorom značajki

$$\langle a_1(x), a_2(x), \ldots, a_n(x) \rangle$$

gdje je *a(x)* označava k-ti atribut primjera *x*.

Euklidska udaljenost između dva vektora x_i i x_j je:

$$d(x_i, x_j) \equiv \sqrt{\sum_{r=1}^{n} \left(a_r(x_i) - a_r(x_j)\right)^2} \qquad [11]$$

Za ciljnu funkciju s diskretnim vrijednostima vrijedi:

$f : R_n \to V$

gdje je $V = \{v_1, v_2, \ldots v_s\}$.

Za ciljnu funkciju s realnim vrijednostima $f : R_n \to R$ se umjesto najčešće pojavljivanje vrijednosti ciljne funkcije uzima srednja vrijednost ciljnih funkcija *k* najbližih susjeda.

$$f(x_q) \leftarrow \frac{\sum_{i=1}^{k} f(x_i)}{k} \qquad [12]$$

Decizijska funkcija vraća pripadnost grupi primjera kojoj pripada najviše od k susjeda testnog primjera, ako *je k=1*, testni primjer se klasificira kao i njegov najbliži susjed. U praksi je *k* često neparan broj da bi se izbjegli slučajevi da jednak broj susjeda pripada u više grupa.

4.1 Algoritam k-najbližih susjeda

Algoritam za učenje:

- Za svaki primjer za učenje *(x,f(x))* dodaj primjer na listu primjeri_za_učenje

Algoritam klasifikacije:

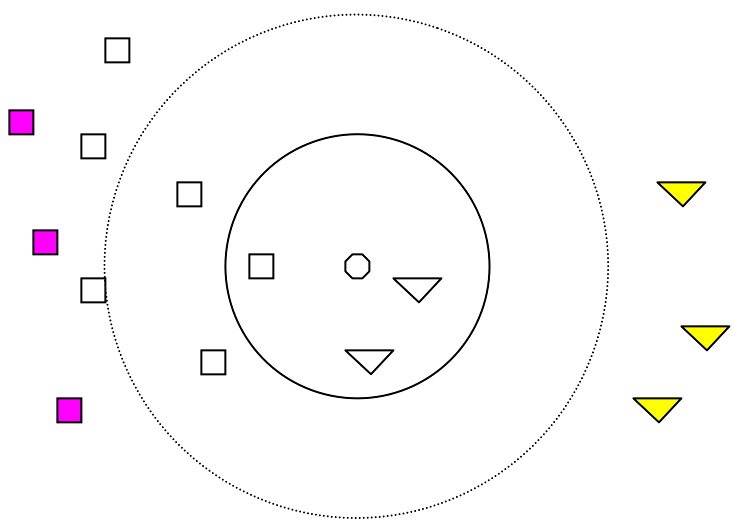
- Za dani primjer x_q s nepoznatom klasifikacijom
    - Neka $x_1, x_2, .. x_k$ označavaju *k* primjera koji su najbliži x_q
    - Vrati

$$f(x_q) \leftarrow \arg\max_{v \in V} \sum_{i=1}^{k} \delta(v, f(x_i)) \qquad [13]$$

$f(x_q)$ je najčešća vrijednost ciljne funkcije koja se pojavljuje među *k* primjera za učenje koji su najbliži primjeru x_q.

Primjer rada algoritma

Testni primjer se u zavisnosti od odabranog *k* različito klasificira. Zajedno s trokutima za *k=3* i s kvadratima za *k = 5*.



Slika 1. Primjer klasifikacije knn algoritmom za k=3 i k=5

4.2 Modifikacija algoritma k–najbližih susjeda uvođenjem težinskih faktora

Budući da je algoritam radi na intuitivnoj pretpostavci da su objekti s najmanjom udaljenosti potencijalno slični, poboljšanje algoritma koje se samo nameće je uvođenje težinskih faktora. Za svaki od k susjeda, uvodi se težinski faktor w_i koji ovisi o njegovoj udaljenosti od upita x_q.

Vrijednost decizijske funkcije određuje se formulom:

$$f(x_q) \leftarrow \arg\max_{v \in V} \sum_{i=1}^{k} w_i \delta(v, f(x_i)) \qquad [14]$$

gdje je

$$w_i \equiv \frac{1}{d(x_q, x_i)^2} \qquad [15]$$

Ako se testni primjer poklapa s primjerom za učenje, tj udaljenost između njih je jednaka nuli, testni primjer se klasificira jednako kao taj primjer za učenje. Ukoliko je više primjera koji se preklapaju, uzima se klasifikacija većine primjera.

Modifikacija za slučaj kontinuirane ciljne funkcije:

$$f(x_q) \leftarrow \frac{\sum_{i=1}^{k} w_i f(x_i)}{\sum_{i=1}^{k} w_i} \qquad [16]$$

gdje je težinski faktor w_i definiran kao i za prethodni slučaj.

Zbog uvođenja težinskih algoritama udaljeni primjeri će imati vrlo malo utjecaja na klasifikaciju. Metode kod kojih se svi primjeri uzimaju u obzir kod klasifikacije, nazivaju se globalnim. Ako se uzima ograničeni broj primjera, onda je to lokalna metoda. Ovakva globalna metoda naziva se Shepardova metoda.

4.3 Karakteristike algoritma k – najbližih susjeda

Prednosti algoritma su robusnost na šum u primjerima za učenje i efikasnost ako je skup primjera za učenje dovoljno velik. Glavni nedostatak je induktivna pristranost, pretpostavlja se da je klasifikacija upita slična klasifikaciji primjera u blizini. Udaljenost se računa na temelju svih atributa, što dovodi do kletve dimenzionalnosti (eng. *curse of dimensionality*), tj. osjetljivosti algoritma na sve atribute, bez obzira na njihov broj i značaj za ciljnu funkciju. Rješenje je množenje atributa s faktorima da bi se smanjio utjecaj nevažnih atributa. Drastičniji pristup je potpuno uklanjanje nevažnih atributa. Također je problem velika složenost izvođenja, zbog toga je potrebno ostvariti efikasno indeksiranje memorije.

5 Stabla odluke

Stabla odluke su metoda za aproksimiranje funkcije diskretnih vrijednosti. Robusna su na šum i mogu učiti i disjunktne izraze. Jedna su od najčešćih i najpraktičnijih metoda induktivnog zaključivanja. Induktivna pristranost stabla odluka je u preferiranju malih stabala u odnosu na velika. Stabla odluke pretražuju potpun prostor hipoteza.

5.1 Problemi prikladni za rješavanje pomoću stabla odluke

Stabla odluke su najbolja za rješavanje problema sa sljedećim karakteristikama:

- Primjeri su predstavljeni parovima atribut-vrijednost, pogotovo ako je skup vrijednosti koje može poprimit atribut malen, moguće je prilagoditi algoritam da radi i sa kontinuiranim vrijednostima atributa.
- Ciljna funkcija poprima diskretne vrijednosti, moguće je prilagoditi algoritam da radi i s realnim vrijednostima.
- Rješenje problema zahtjeva disjunktni izraz. Stabla odluke prirodno predstavljaju disjunktni izraz.
- Primjeri za učenje sadržavaju pogreške. Stabla odluke su robusna i na pogreške u klasifikaciji primjera za učenje i na pogreške u vrijednostima atributa primjera za učenje.
- Primjerima za učenje nedostaju neke vrijednosti atributa. Stabla odluke se mogu koristiti ako postoje nepoznate vrijednosti atributa.

Naučene funkcije su predstavljene kao stabla odluke ili kao skup ako-onda pravila radi čitljivosti i preglednosti.

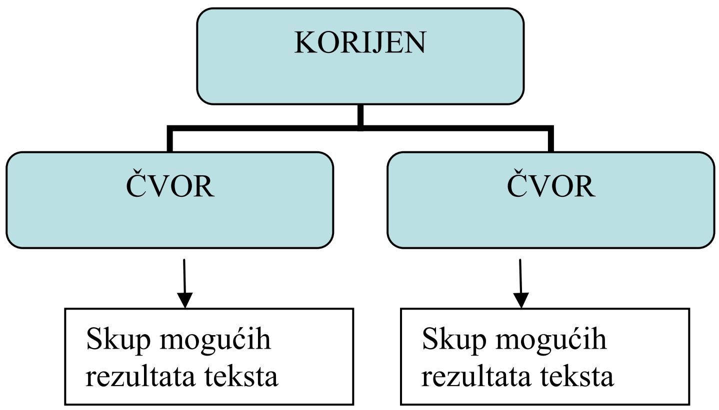
5.2 Predstavljanje stabla odluke

Stablo odluke predstavlja disjunkciju konjunkcije uvjeta na vrijednosti atributa.. Svaki put od korijena stabla do lista korespondira s konjunkcijom vrijednosti atributa, a stablo je disjunkcija ovih konjunkcija.

Klasifikacija primjera se vrši odozgo, od korijena prema listovima. Svaki čvor na stablu predstavlja testiranje određenog atributa primjera, a svaka grana koja izlazi iz tog čvora predstavlja jednu od vrijednosti za taj atribut.

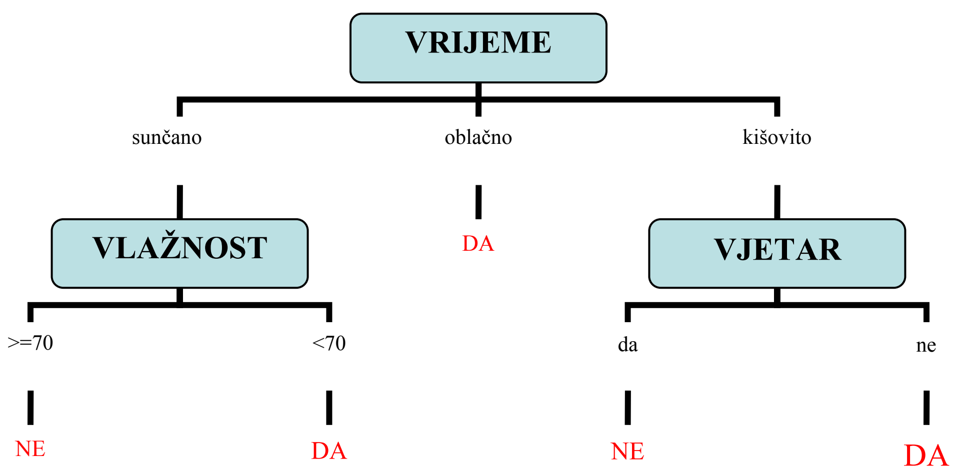
Dijagram

- Svaki čvor koji nije list je povezan sa testom koji skup mogućih vrijednosti atributa dijeli na podskup koji odgovara različitim rezultatima testa.
- Svaka grana prenosi rezultate određenog testa na sljedeći čvor
- Svaki čvor je povezan sa skupom mogućih rezultata testa
- Za m atributa stablo odluke smije imati visinu manju ili jednaku m.



Slika 2. Dijagram stabla odluke

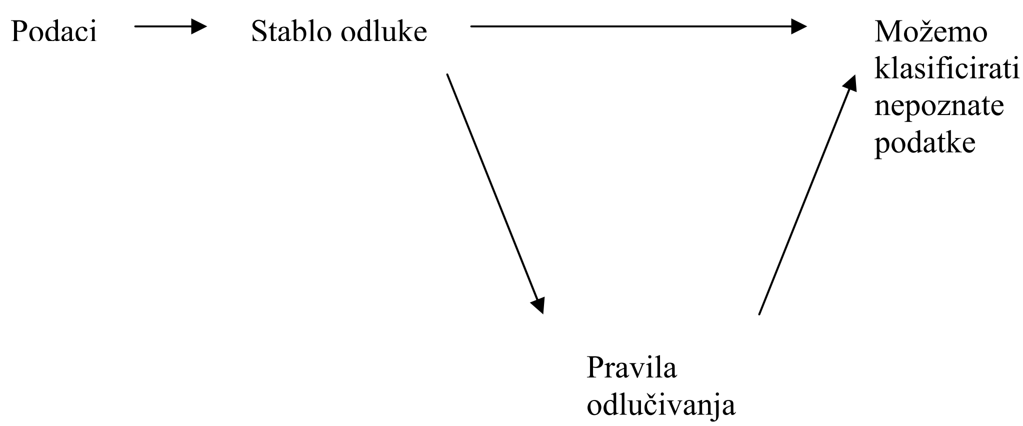
Primjer : Da li je vrijeme pogodno za golf?



Slika 3. Primjer stabla odluke

5.3 Osnovni algoritam učenja stabla odluke

Stablo odluke se konstruira promatrajući pravilnosti u podacima :



Slika 4. Rad stabla odluke

Temeljni algoritam učenja stabla odluke je ID3 (eng. *Induction of Decision Trees)*, proširenje tog algoritma je C4.5.

ID3:

- Testira se svaki atribut da se ocjeni kako dobro klasificira primjere
- Najbolji se odabire kao čvor, a njegove vrijednosti su silazne grane
- Primjeri za učenje se sortiraju prema odgovarajućem silaznom čvoru (niz onu granu koja odgovara vrijednosti tog atributa)
- Cijeli postupak se ponavlja koristeći primjere koji su dodijeljeni silaznom čvoru

ID3 spada u pohlepne algoritme (eng. *greedy algorithm*), jer se nikad ne vraća zbog ponovnog razmatranja prethodnih čvorova.

Potrebno je odabrati koji atribut će se testirati u pojedinom čvoru stabla.

Informacijska dobit (eng. *information gain*) je mjera kako dobro pojedini atribut dodjeljuje primjere za učenje u skladu s ciljnom klasifikacijom. Informacijsku dobit računamo pomoću entropiju, koja mjeri homogenost primjera.

Ako je:

n_b - broj primjera u grani b

n_{bc} - broj primjera u grani b klase c, naravno n_{bc} je jednaki ili manji od n_b

n_t - ukupni broj primjera u svim granama

Vjerojatnost:

P_b - vjerojatnost da je primjer u grani b pozitivan

$$P_b = \frac{broj\_pozitivnih\_primjera\_na\_grani}{ukupni\_broj\_primjera\_na\_grani} = \frac{n_{bc}}{n_b}$$

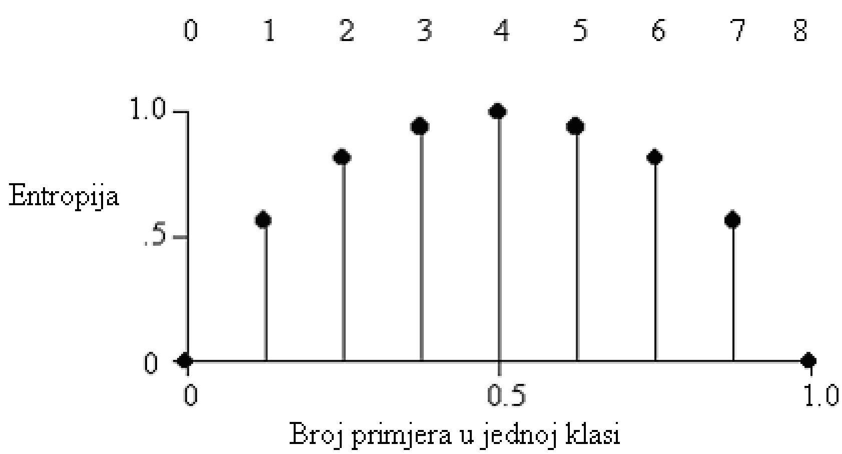
- Ako su svi primjeri na grani pozitivni, onda je *Pb = 1* (homogeno pozitivan)
- Ako su svi primjeri na grani negativni, onda je *Pb = 0* (homogeno negativan)

Entropija:

$$Entropija = \sum_{C} -(\frac{n_{bc}}{n_b})\log_2(\frac{n_{bc}}{n_b}) \qquad [17]$$

- Entropija varira između 0 i 1
- *Entropija = 0* ako je skup potpuno homogen
- *Entropija = 1* ako je skup potpuno nehomogen

Primjer entropije za skup od 6 primjera koji mogu pripadati u dvije klase:



Slika 5. Primjer entropije

Srednja entropija:

$$Srednja\_entropija = \sum_{b}(\frac{n_b}{n_t}) \times \left[\sum_{C} -(\frac{n_{bc}}{n_b})\log_2(\frac{n_{bc}}{n_b}) \right] \quad [18]$$

Informacijska dobit primjera A u odnosu na skup primjera S:

$$Informacijska\_dobit(S, A) = Entropija(S) - \sum_{v \in Vrijednost(A)} \frac{|S_v|}{|S|} Entripija(S_v) \quad [19]$$

Vrijednost(A) – skup svih mogućih vrijednosti skupa *A*

Sv – podskup od *S* za koji atribut *A* ima vrijednost v, tj. $S = \{s \in S | A(s) = v\}$

Za čvor u stablu biramo atribut s najvećom informacijskom dobiti i nastavljamo dalje s algoritmom.

5.4 Karakteristike algoritma ID3

Algoritam ID3 je induktivno pristran na dva načina, izabire kraće stablo prije nego dulje stablo i atribute s većom informacijskom dobiti stavlja bliže korijenu stabla.

Može doći do prenaučenosti (eng. *overfit*) stabla odluke.

Definicija prenaučenosti:

Neka je dan prostor hipoteza *H*. Hipoteza $h \in H$ je prenaučena ako postoji hipoteza $h' \in H$ takva da:

h ima manju pogrešku nego *h'* na primjerima na učenje, ali *h'* ima manju pogrešku nego *h* na cijelom prostoru primjera.

Prenaučenost se rješava zaustavljanjem rasta stabla prije savršene klasifikacije i naknadnim podrezivanjem stabla, što je i bolji pristup.

6 Naivni Bayesov klasifikator

Probabilistički pristup indukciji znanja temelji se na indukciji znanja Bayesovim učenjem. Osnovna ideja je da se vrijednosti promatranih atributa instanci ravnaju po određenim razdiobama vjerojatnosti, te da se optimalno zaključivanje o novoj instanci može izvesti iz tih vjerojatnosti primijenjenih na njene atribute.

6.1 Bayesov teorem

Bayesov teorem se temelji na odabiru najvjerojatnije hipoteze iz skupa hipoteza *H* na osnovu skupa za učenje *D*, a uz utjecaj predodređenih vjerojatnosti svake od ponuđenih hipoteza u skupu *H*.
Da bi objasnili teorem potrebno je definirati vjerojatnosti:

- *P(h)* - a priori vjerojatnost hipoteze . Apriorna vjerojatnost hipoteze može biti "objektivna" kada se temelji na stvarnom eksperimentu ili "subjektivna" kada se temelji na pretpostavci. Ukoliko vrijednost vjerojatnosti nije poznata može se svim hipotezama pridijeliti jednaka početna vjerojatnost.
- *P(D)* - vjerojatnost pojavljivanja instance *D*.
- *P(D|h)* - uvjetna vjerojatnost pojavljivanja instance i ako je hipoteza h točna.
- *P(h|D)* - posteriorna vjerojatnost točnosti hipoteze *h* nakon pojavljivanja instance *D*. *P(h|D)* omogućava procjenu točnosti hipoteza nakon promatranja pojave novih instanci *D*, za razliku od a priori vjerojatnosti *p(h)*, koja je neovisna o pojavi podatka *D*.

Formula Bayesovog teorema glasi:

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)} \qquad [20]$$

Najčešće je potrebno izračunati maksimalnu aposteriornu hipotezu (MAP) $h \in H$, hipotezu s najvećom vjerojatnošću h_{MAP} nakon što se dogodio određeni događaj *D*:

$$\begin{aligned} h_{MAP} &= \arg\max_{h\in H} P(h \mid D) \\ &= \arg\max_{h\in H} \frac{P(D \mid h)P(h)}{P(D)} \\ &= \arg\max_{h\in H} P(D \mid h)P(h) \end{aligned} \qquad [21]$$

U posljednjem retku je izbačen *P(D)* iz nazivnika jer on ne ovisi o hipotezi *h*.

U slučajevima kada a priori vjerojatnosti hipoteza *h* jednake, možemo zanemariti utjecaj parametra *P(h)* i procjenjujemo samo na osnovu *P(D|h)*. Veličina h_{MAP} se pojednostavljuje na slijedeći način:

$$h_{MAP} = \arg\max_{h \in H} P(D \mid h) \qquad [22]$$

Premda ovakav klasifikator dokazano polučuje najbolju klasifikaciju, problematičan je i neprimjenjiv u velikom broju slučajeva zbog vrlo velikog broja trening instanci potrebnih kako bi se izračunale uvjetne vjerojatnosti kojima se barata. Stoga se u praktičnoj primjeni koristi pojednostavljeni oblik Bayesovog klasifikatora nazvan Naivni Bayesov klasifikator.

6.2 Naivni Bayesov klasifikator – teorija

Naivni Bayesov klasifikator se primjenjuje u slučajevima gdje se primjer podatka za učenje može prikazati kao konjunkcija atributa koji mogu poprimiti određeni (konačan) skup vrijednosti (a_1, a_2, ... ,a_n). Uvodimo pojam najvjerojatnije klasifikacije v_{MAP} koji predstavlja najvjerojatniji element konačnog skupa *V* svih mogućih klasifikacija ulazne instance i može se računati po izrazu:

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j \mid a_1, a_2 ... a_n) \qquad [23]$$

Prethodni izraz možemo raspisati po Bayesovom teoremu:

$$v_{MAP} = \arg\max_{v_j \in V} \frac{P(a_1, a_2 ... a_n \mid v_j) P(v_j)}{P(a_1, a_2 ... a_n)} \qquad [24]$$

$$v_{MAP} = \arg\max_{v_j \in V} P(a_1, a_2 ... a_n \mid v_j) P(v_j) \qquad [25]$$

Vjerojatnost $P(v_j)$ nije problem izračunati jednostavnim pobrojavanjem pojavljivanja tražene rezultantne klasifikacije. Problem dolaze s izračunom izraza $P(a_1, a_2 ... a_n|v_j)$ zbog međusobne zavisnosti vrijednosti atributa $a_1, a_2 .... a_n$ tako da je broj mogućih $P(a_1, a_2 ... a_n|v_j)$ izraza jednak broju svih mogućih različitih n-torki pomnoženih sa brojem svih mogućih klasifikacija. Da bi se odredile $P(a_1, a_2 ... a_n|v_j)$ za sve moguće kombinacije vrijednosti atributa potrebna je ogromna količina podataka u setu za učenje.
Stoga se uvodi daljnje pojednostavljenje koje se bazira na tzv. "naivnoj pretpostavci" da su pojave vrijednosti različitih atributa međusobno nezavisne:

$$P(a_1, a_2 ... a_n \mid v_j) = \prod_i P(a_i \mid v_j) \qquad [26]$$

Primjenom Bayesovog teorem dobivamo formulu:

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i \mid v_j) \qquad [27]$$

Broj različitih vjerojatnosti $P(a_i|v_j)$ koje treba izračunati iz podataka sada je mnogo manji broj nego broj potreban kako bi se dobila $P(a_1, a_2 ... a_n|v_j)$.

6.3 Primjena Naivnog Bayesovog klasifikatora u klasifikaciji teksta

U praktičnoj upotrebi Naivni Bayesov klasifikator je pokazao korisnost zbog jednostavnosti implementacije i polučenih zadovoljavajućih rezultata. Jedan od najčešćih primjena je klasifikacija odnosno filtriranje e–mail poruka (anti–spam softver).

Primjer klasifikacije dokumenata na osnovu sadržaja

Pregledavanjem sadržaja poruka izdvajamo atribute w_i (u našem slučaju riječi) gdje se vjerojatnost da će se riječ w_i određenog dokumenta svrstati u klasifikaciju v_j može se napisati kao $P(a_i \mid v_j)$. Pretpostavljanjem međusobne nezavisnosti atributa dolazimo do izraza vjerojatnosti dokumenta *D:*

$$P(D \mid v_j) = \prod_i P(w_i \mid v_j) \qquad [28]$$

Prema teoriji vjerojatnosti gornji izraz možemo napisati kao:

$$P(D \mid v_j) = \frac{P(D \cap v_j)}{P(v_j)} \qquad [29]$$

ili

$$P(v_j \mid D) = \frac{P(D \cap v_j)}{P(D)} \qquad [30]$$

Primjenom Bayesovog teorema na gornji izraz dobivamo:

$$P(v_j \mid D) = \frac{P(v_j)}{P(D)} P(D \mid v_j) \qquad [31]$$

Ako klasifikaciju v_j svedemo samo na dvije vrijednosti S i ¬S (spam e-mail poruka i legitimna e-mail poruka) dolazimo do izraza:

$$P(D \mid S) = \prod_i P(w_i \mid S) \qquad [32]$$

odnosno

$$P(D \mid \neg S) = \prod_i P(w_i \mid \neg S) \qquad [33]$$

Koristeći prethodno izvedene izraze možemo napisati:

$$P(S \mid D) = \frac{P(S)}{P(D)} \prod_i P(w_i \mid S) \qquad [34]$$

i

$$P(\neg S \mid D) = \frac{P(\neg S)}{P(D)} \prod_i P(w_i \mid \neg S) \qquad [35]$$

Dijeljenjem dobivamo:

$$\frac{P(S \mid D)}{P(\neg S \mid D)} = \frac{P(S)}{P(\neg S)} \frac{\prod_i P(w_i \mid S)}{\prod_i P(w_i \mid \neg S)} = \frac{P(S)}{P(\neg S)} \prod_i \frac{P(w_i \mid S)}{P(w_i \mid \neg S)} \qquad [36]$$

Budući vrijedi da je p(S | D) + p(¬S | D) = 1, možemo izračunati vjerojatnost *P(S|D)* na slijedeći način:

$$\ln \frac{P(S \mid D)}{P(\neg S \mid D)} = \ln \frac{P(S)}{P(\neg S)} + \sum_i \ln \frac{P(w_i \mid S)}{P(w_i \mid \neg S)} \qquad [37]$$

Dakle, poruka se klasificira kao spam e-mail poruka za $\ln \frac{P(S \mid D)}{P(\neg S \mid D)} > 0$

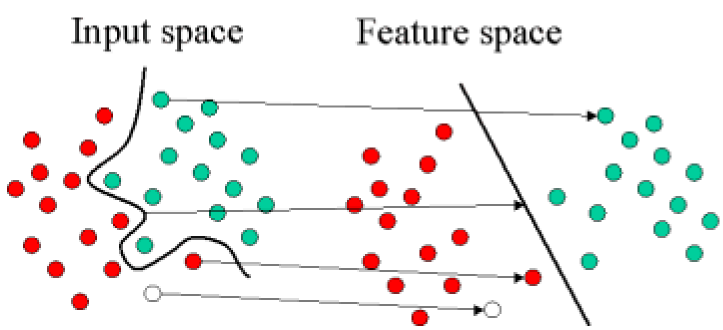
6.4 Karakteristike Naivnog Bayesovog klasifikatora

Dobra strana ove metode je brzina, čak i za velike skupove podataka. U teoriji Bayesov klasifikator u odnosu na druge klasifikatore ima minimalnu pogrešku, međutim u praksi to često nije tako. Najveći nedostatak ove metode je u temeljnoj *naivnoj* pretpostavci da je djelovanje nekog atributa na pripadnost uzorka pojedinom razredu neovisno o djelovanju drugih atributa, odnosno da su atributi međusobno neovisni, što u praksi nije uvijek istina.

7 Metoda potpornih vektora - SVM

Metoda potpornih vektora, odnosno algoritam *SVM* (eng. *Support Vector Machines*) spada u grupu jezgrenih metoda klasifikacije. Temelji se na principu strukturne minimizacije rizika koji pronalazi hipotezu *h* za koju se može garantirati najmanja vjerojatnost pogreške na skupu za učenje.

Metoda potpornih vektora vrši klasifikaciju preslikavanjem skupa pokaznih uzoraka iz ulaznog prostora uzoraka $\Re_N$ (eng. *input space*) u *N*-dimenzionalni prostor *F* (eng. *feature space*) koji optimalno razdvaja uzorke u dvije kategorije. Nakon što se vektor uzorka *x* preslika u prostor *F* funkcijom *Φ*, u novom prostoru se određuje kojoj kategoriji novi vektor *Φ*(x) pripada. Kategorije razdvaja hiperravnina razdvajanja. Optimalna hiperravnina je ona s maksimalnom granicom razdvajanja između uzoraka dvaju razreda.



Slika 6. Preslikavanje pokaznih uzoraka iz ulaznog prostora (eng. *input space*) u *N*-dimenzionalni prostor *F* (eng. *feature space*)

Zadan je skup pokaznih uzoraka *{x1, ..., xl},* koji ima *l* elemenata. Kategorija kojemu pripada *i*-ti uzorak označen je s *y*i i može biti ±1. skup uzoraka se preslikava nelinearnim mapiranjem u N-dimezionalni prostor *F*:

$\Phi : \Re_N \rightarrow F$

Novi uzorci, čija kategorija nije poznata, razvrstavaju se u prostor *F* funkcijom odlučivanja *f* koja glasi:

$$f(\mathbf{x}) = sign((\mathbf{wx}) + b) = \pm 1, \qquad [38]$$

gdje je $(\mathbf{wx}) + b = 0$, $\mathbf{w} \in \Re_N$, $b \in R$ kategorija hiperravnina.

Vektor w jednak je $w = \sum_i v_i x_i$, gdje vektori x_i predstavljaju primjere koji su najbliže hiperravnini, a nazivaju se potporni vektori.

Budući potporni vektori nose informacije o problemu razvrstavanja, funkcija odlučivanja f glasi:

$$f(x) = sign(\sum_i v_i (xx_i) + b) = \pm 1 \quad [39]$$

Sada se u prostoru F računaju se skalarni produkti $k(x,x_i) = \Phi(x)\ \Phi(x_i)$, gdje je k funkcija jezgre, a težine v_i se određuju rješavanjem kvadratnih jednadžbi.
Kategorija uzorka se određuje s:

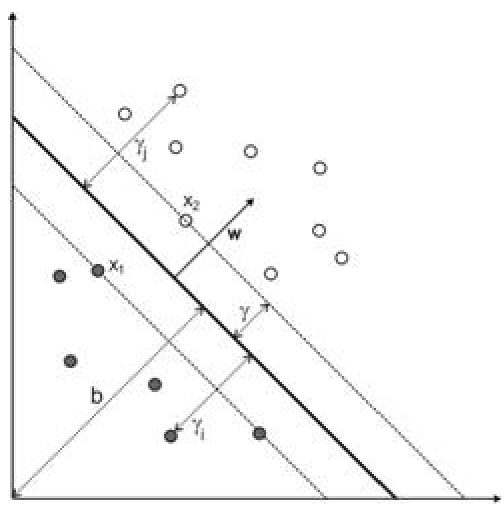
$$f(x) = sign(\sum_i v_i k(xx_i) + b) = \pm 1 \quad [40]$$

7.1 Klasifikator s maksimalnom marginom

Pretpostavimo da je skup pokaznih uzoraka, odnosno skup za učenje linearno razdvojiv, tj. da postoji hiperravnina (**w**,b) tako da za sve primjere skupa vrijedi:

$$y_i(\langle w \mid x_i \rangle + b) > 0 \quad [41]$$

Tada postoji više hiperravnina koje razdvajaju skup za učenje bez pogreške.



Slika 7. Klasifikator s maksimalnom marginom

Na slici 5.2 je prikazana hiperravnina gdje γ predstavlja maksimalnu marginu, a primjeri x_i potporne vektore. Skaliranjem hiperravnine (**w**,b) pomoću skalara $\alpha \in R+$ u (λ**w**, λb), ne mijenja se funkcija vezana uz hiperravninu. Stoga možemo skalirati parametre hiperravnine tako da funkcijska margina iznosi 1. Dakle slijedi:

$$\langle w \mid x_2 \rangle + b = +1$$ [42]
$$\langle w \mid x_1 \rangle + b = -1$$

$$\langle w \mid x_2 - x_1 \rangle = 2$$ [43]

$$\gamma = \frac{1}{2}\left\langle \frac{w}{\|w\|} \mid x_2 - x_1 \right\rangle$$ [44]

$$\gamma = \frac{1}{\|w\|}$$

Margina γ je obrnuto proporcionalna s udaljenosti između bilo koja dva primjera različitih kategorija., stoga je za maksimalnu marginu potrebno pronaći:

$$\min\left[\frac{1}{2}\|w^2\|\right]$$ [45]

uz zadovoljene uvjete:

$$y_i\left(\langle w \mid x_i \rangle + b\right) \geq 1, \quad i = 1...l$$ [46]

Budući je je zadano ograničenje funkcije, odnosno uvjet [46], izraz [45] nije lako numerički rješiti. Problem se rješava uporabom Lagrangeovih multiplikatora. Neka je *f(x,y)* funkcija čije je ekstreme potrebno naći, a neka je $g(x, y) = c$ uvjet.. U točkama ekstrema gradijenti funkcija *f* i *g* su paralelni vektori normala, te je zadovoljen slijedeći izraz:

$$\nabla f(x, y) = \alpha \ \nabla g(x, y)$$ [47]

gdje je α Lagrangeov multiplikator[1], pomoću kojeg izjednačavamo funcije *f* i *g* po duljini. Pomoći izraza [47] i uvjeta $g(x, y) = c$ dolazimo do Lagrangianove formule:

$$L(x, y, \alpha) = f(x, y) + \alpha\left(g(x, y) + c\right)$$ [48]

a točke ekstrema dobivamo pomoću:

$$\nabla L(x, y, \alpha) = 0$$ [49]

U slučaju kada imamo više uvjeta $g_i(x, y)$, problem pronalaska ekstrema se rješava na slijedeći način:

[1] Optimizacijski postupci, a među njima i teorija Lagrangiana opisani su u (Cristianini, 2000) [4]

$$\nabla f(x, y) = \sum_i \alpha_i \nabla g_i(x, y) \qquad [50]$$

odnosno

$$L(x, y, \alpha) = f(x, y) - \sum_i \alpha (g(x, y) + c) \qquad [51]$$

Uvrštavanjem izraza [45] i [46] u prethodnu funkciju dobivamo dualnu formu:

$$L(w, b, \alpha) = \frac{1}{2}\langle w \mid w\rangle - \sum_{i=1}^{l} \alpha_i \left[y_i \left(\langle w \mid x_i \rangle + b \right) - 1 \right] \qquad [52]$$

Derivacije od *L* po primalnim varijablama u sedlu moraju nestati, pa slijedi:

$$\frac{\partial L(w, b, \alpha)}{\partial w} = w - \sum_{i=1}^{l} y_i \alpha_i x_i = 0 \qquad [53]$$

$$\frac{\partial L(w, b, \alpha)}{\partial b} = \sum_{i=1}^{l} y_i \alpha_i = 0 \qquad [54]$$

Transformacijama ovih izraza dobivamo:

$$w = \sum_{i=1}^{l} y_i \alpha_i x_i \qquad [55]$$

$$0 = \sum_{i=1}^{l} y_i \alpha_i \qquad [56]$$

Uvrštavanjem u Lagrangian dobivamo dualnu formu koju je potrebno maksimizirati.

$$\begin{aligned} L(w, b, \alpha) &= \frac{1}{2}\langle w \mid w\rangle - \sum_{i=1}^{l} \alpha_i \left[y_i \left(\langle w \mid x_i \rangle + b \right) - 1 \right] \\ &= \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \langle x_i \mid x_j \rangle - \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \langle x_i \mid x_j \rangle + \sum_{i=1}^{l} \alpha_i \qquad [57] \\ &= \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \langle x_i \mid x_j \rangle \end{aligned}$$

$$\max\left[W(\alpha)=\sum_{i=1}^{l}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{l}y_iy_j\alpha_i\alpha_j\langle x_i \mid x_j\rangle\right]$$

$$\sum_{i=1}^{l}y_i\alpha_i=0 \qquad [58]$$

$$\alpha_i\geq 0\,,\quad i=1..l$$

Neka je α^* rješenje dualne forme, tada težinski vektor w^* daje hiperravninu s maksimalnom marginom:

$$w^*=\sum_{i=1}^{l}y_i\alpha_i^*x_i \qquad [59]$$

$$\gamma=\frac{1}{\|w^*\|} \qquad [60]$$

Odmak b je potrebno izračunati iz primalnih varijabli:

$$b^*=-\frac{\max_{y_i=-1}\left(\langle w^* \mid x_i\rangle\right)+\min_{y_i=+1}\left(\langle w^* \mid x_i\rangle\right)}{2} \qquad [61]$$

Prema Kuhn-Tuckerovom teoremu postoji i dopunski uvjet

$$\alpha_i\left[y_i\left(\langle x_i \mid w\rangle+b\right)-1\right]=0\,,\quad i=1..l \qquad [62]$$

pomoću kojeg zaključujemo kako su jedino za one primjere x_i, za koje funkcijska margina iznosi 1, te leže točno na geometrijskoj margini odgovarajući α_i^* različiti od nule. Ostali primjeri su nevažni.

7.2 Metoda potpornih vektora sa slabom marginom

Prethodno opisana metoda potpornih vektora s maksimalnom marginom pretpostavlja da su primjeri za učenje linearno razdvojivi. Međutim, ukoliko skup za učenje sadrži šum, primjeri neće biti linearno razdvojivi, pa je potrebno uvesti metodu potpornih vektora sa slabom marginom.

Uvodimo varijable *ξi* koje dopuštaju primjerima da stoje izvan granica margine te da budu krivo klasificirani.

$$y_i\left(\langle w \mid x_i\rangle+b\right)\geq 1-\xi_i\,,\quad i=1..l \qquad [63]$$

$$\xi_i\geq 0\,,\quad i=1..l$$

Da bi se pogreške što više smanjile, koristimo slijedeće metode: slaba margina u L2 normi

$$\min\left[\frac{1}{2}\left\|w^2\right\| + \frac{1}{2}C\sum_{i=1}^{l}\xi_i^2\right] \qquad [64]$$

$$y_i\left(\left\langle w \mid x_i \right\rangle + b\right) \geq 1 - \xi_i, \quad i = 1...l$$

te slaba margina u L1 normi.

$$\min\left[\frac{1}{2}\left\|w^2\right\| + \frac{1}{2}C\sum_{i=1}^{l}\xi_i\right] \qquad [65]$$

$$y_i\left(\left\langle w \mid x_i \right\rangle + b\right) \geq 1 - \xi_i, \quad i = 1...l$$

$$\xi_i \geq 0, \quad i = 1...l$$

Optimalna vrijednost parametra C je apriori nepoznata te se određuje krosvalidacijom na skupu za učenje.

Nakon nekoliko koraka ekvivalentnih onima za klasifikaciju s maksimalnom marginom dobivamo dualne definicije problema:

$$\max\left[W(\alpha) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \left(\left\langle x_i \mid x_j \right\rangle + \frac{1}{C}\delta_{ij}\right)\right] \qquad [66]$$

$$\sum_{i=1}^{l} y_i \alpha_j = 0,$$

$$\alpha_i \geq 0, \quad i = 1...l$$

$$\max\left[W(\alpha) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \left\langle x_i \mid x_j \right\rangle\right] \qquad [67]$$

$$\sum_{i=1}^{l} y_i \alpha_j = 0$$

$$C \geq \alpha_i \geq 0, \quad i = 1...l$$

7.3 Karakteristike metodu potpornih vektora (SVM)

Metoda potpornih vektora nalazi primjenu u kategorizaciji teksta, prepoznavanju rukopisa, klasifikaciji slika, itd. Prilagođena je za baratanje velikim količinama podataka – štoviše, mnoštvo mjerenja povećava mogućnost razlučivanja pouzdano opaženih i bitnih uzoraka u podacima, od onih nepouzdanih, ili nebitnih. Popularna je zbog maksimalne generalizacije. Prilično je otporna na šum u podacima, neizbježnu posljedicu bilo kojeg eksperimentalnog mjerenja. Međutim točnost metode potpornih vektora je ograničena i popriličnoovisi o izboru nekih parametrima kao što su C, γ, itd. Također, metoda je predviđena za klasifikaciju u samo dvije kategorije, ali razvijeni su i različiti pristupi koji omogućuju klasifikaciju više od dvije kategorije.

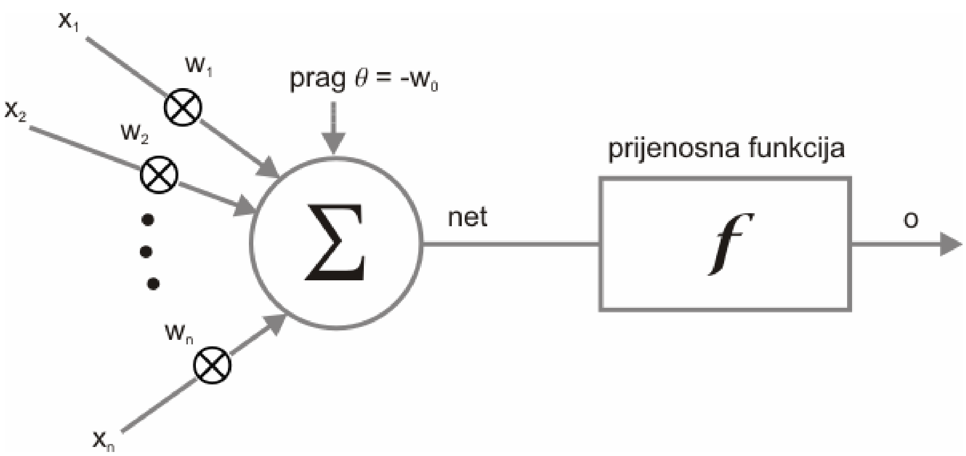
8 Umjetne neuronske mreže

Umjetna neuronska mreža je matematički ili računalni model temeljen na biološkoj neuronskoj mreži. Sastoji se od više međusobno povezanih umjetnih neurona i procesnih informacija koji služe distribuiranoj paralelnoj obradi podataka. Umjetne neuronske mreže uče pomoću primjera.

Dobro rješavaju probleme klasifikacije i predviđanja. Mogu raditi sa složenim i nepotpunim podacima. Neke od ostalih prednosti neuronskih mreža su prilagodljivost okolini, samoorganizacija i vršenje operacija u realnom vremenu.

8.1 Model umjetne neuronske mreže

Umjetni neuron se sastoji od više ulaza i jednog izlaza. Svaki ulaz x_i ima pridruženu težinu w_i, izlaz o je kompozicija ulaza pomnoženih sa odgovarajućim težinama.



Slika 8. Slojevi umjetne neuronske mreže

Vrijedi:

$$net = \sum_{i=1}^{n} w_i x_i + \theta \qquad [68]$$

$$\theta = -w_0$$

$$x_0 = 1$$

$$net = \sum_{i=0}^{n} w_i x_i \; [69]$$

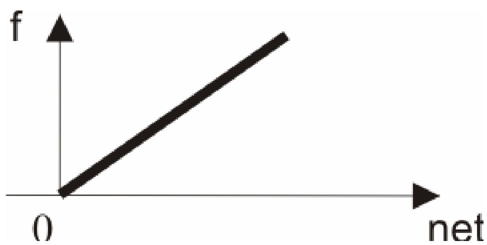
$$o = f\left(\sum_{i=0}^{n} w_i x_i\right) = f(net)$$

Prihvatljiva razina izlaza je obično između 0 i 1, ili između -1 i 1.

Prijenosna funkcija f je predefinirana, obično se koriste funkcije:

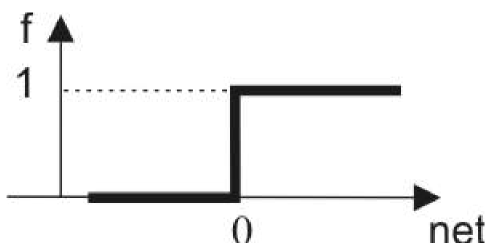
- ADALINE

  $f(net) = net$



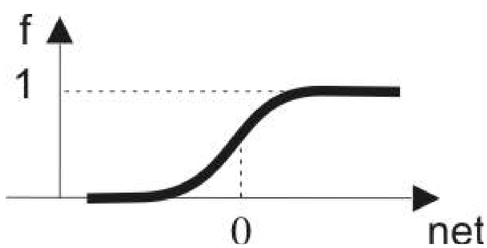
- TLU

  $f(net) = \begin{cases} 0\_za\_net<0 \\ 1\_inačn \end{cases}$



- Sigmoidalna funkcija

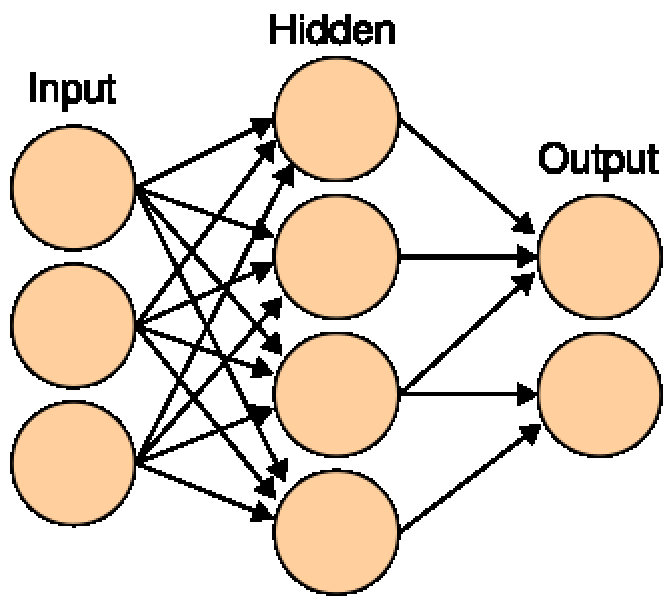
  $f(net) = \dfrac{1}{1+e^{-o*net}}$



8.2 Topologija neuronskih mreža

Projektiranje neuronske mreže se vrši određivanjem broja neurona i njihovim povezivanjem, tj definiranjem njene topologije. Razlikujemo dvije vrste neuronskih mreža prema njihovoj topologiji. To su feed-forward neuronske mreže u kojima nema povratne veze, signal se prostire samo u jednom smjeru, ne vraća se u niže slojeve mreže i neuronske mreže s povratnom vezom. Neuronske mreže s povratnom vezom su složenije i imaju veće sposobnosti dinamičkog procesiranja od feed-forward mreža.

Neuronsku mrežu možemo podijeliti u više neuronskih slojeva. Svaki sloj prima podatke iz prethodnog sloja i šalje sljedećem sloju. Obično se razlikuju tri sloja, ulaz, izlaz i skriveni sloj (eng *hidden layer*). Na ulaz se šalju ulazni podaci, u skrivenom sloju se obrađuju, a na izlazu se dobivaju rezultati te obrade. Mreža se ponaša kao crna kutija, poznati su sami ulaz i izlaz, dok je obrada skrivena.



Slika 9. Topologija neuronskih mreža

8.3 Učenje

Neuron ima dva načina ili faze rada, faza učenja i faza obrade podataka. U fazi učenja potrebno je prvo definirati težine pojedinih ulaza, nakon toga se težine pojedinih ulaza korigiraju prema testnim primjerima.

Razlikuju se dva načina učenja:

- Učenje s učiteljem (eng *supervised learning*) – Učenje se odvija uz skup primjera.
- Učenje bez učitelja (eng *unsupervised learning*) – Ne postoji skup primjera, mogući izlazi nisu odmah poznati.

Učenje se odvija dok mreža dovoljno točno ne obradi podatke.

Najčešći algoritam za učenje neuronske mreže je back-propagation algoritam. Algoritam uspoređuje izlaz neuronske mreže s željenim izlazom i računa greške za svaki čvor u mreži. Težine veza se podešavaju prema vrijednosti greške za svaki čvor dok se ne dobiju minimalne prosječne kvadratne greške. Greške se smanjuju prolaskom kroz mrežu od izlaza prema ulazu.

9 Testiranje

U radu *Text Categorization with Support Vector Machines: Learning with Many Relevant Features* Thorstena Joachima [1] dana je usporedba rezultata rada pet različitih metoda strojnog učenja, metoda potpornih vektora - SVM, Bayesov algoritam, Rocchio algoritam, metoda k-najbližih susjeda s težinskim faktorima i algoritam C4.5 stabla za učenje. Ispitivanje je vršeno na dva testna skupa, prva je „ModApte“ koja je dio Reuters 21578 testnog skupa. Drugi testni skup je preuzet iz Ohsumed[2] kolekcije koju je sastavio William Hersh. Eksperimentalni rezultati su pokazali da se najlošije ponaša Bayes, slijede ga Rocchio algoritam i algoritma C4.5 stabla za učenje. Najbolje rezultate su imali algoritam k-najbližih susjeda i SVM, od ta dva SVM se pokazao kao bolji.

U radu *Machine Learning in Automated Text Categorization* Fabrizia Sebastianija (2002) [2] dan je tablični prikaz uspješnosti različitih vrsta klasifikatora na pet varijanti Reuters testnog skupa dokumenata: Reuters-22173 “ModLewis” (stupac #1), Reuters-22173 “ModApt´e” (stupac #2), Reuters-22173 “ModWiener” (stupac #3), Reuters-21578 “ModApt´e” (stupac #4), i Reuters-21578[3] “ModApt´e” (stupac #5). Način na koji je autor vršio uspoređivanje je objašnjen u nastavku. Da bi se korektno moglo uspoređivati različite klasifikatore trebalo bi primijeniti jednu od dvije metode za uspoređivanje klasifikatora:

- Izravna usporedba: klasifikatori Φ' i Φ'' mogu se uspoređivati samo ako su testirani na istom testnom skupu Ω, obično od strane istih istraživača i u istim uvjetima. Ovo je pouzdanija metoda.
- Neizravna usporedba: klasifikatori Φ' i Φ'' mogu se uspoređivati ako:
    1. testirani su na istim testnim skupovima Ω' i Ω'' respektivno, obično od strane različitih istraživača i prema tome u različitim uvjetima
    2. jedan ili više standardnih klasifikatora $\overline{\Phi_1},...,\overline{\Phi_m}$ su testirani izravnom usporedbom na Ω' i Ω''

    Ova metoda je manje pouzdana.

Test 2 indicira relativnu „težinu“ testnih skupova Ω' i Ω'', koristeći to i rezultate testa 1 može se procijeniti relativna efikasnost klasifikatora Φ' i Φ''.

[2] Ohsumed kolekcija je dostupna na adresi ftp://medir.ohsu.edu/pub/ohsumed

[3] Reuters-21578 kolekcija dostupna je na adresi http://www.research.att.com/~lewis/reuters21578.html.

Tablica 2. Usporedni rezultati različitih klasifikatora ispitanih na pet različitih verzija baza novinskih članaka Reuters. Oznaka "F1" označava uporabu F1 mjere učinkovitosti (van Rijsbergen, 1972, 1979 [8]; Lewis, 1995 [9]), oznaka "M" označava makro usrednjavanje (eng. *macroaverage*)

			#1	#2	#3	#4	#5
		broj dokumenata	21,450	14,347	13,27	12,902	12,902
		broj dok. u skupu za učenje	14,704	10,667	9,610	9,603	9,603
		broj dok. za testiranje	6,746	3,680	3,662	3,299	3,299
		broj kategorija	135	93	92	90	10
Metoda	Vrsta						
WORD		[Yang 1999]	.150	.310	.290		
	probabilistička	[Dumais et al. 1998]				.752	.815
	probabilistička	[Joachims 1998]				.720	
PropBayes	probabilistička	[Lam et al. 1997]	.443 (M*F*1)				
Bim	probabilistička	[Lewis 1992a]	.650				
Nb	probabilistička	[Li and Yamanishi 1999]				.747	
	probabilistička	[Li and Yamanishi 1999]				.773	
	probabilistička	[Yang and Liu 1999]				.795	
	stabla odluke	[Dumais et al. 1998]					.884
C4.5	stabla odluke	[Joachims 1998]				.794	
Ind	stabla odluke	Lewis and Ringuette 1994]	.670				
Swap-1	skupni linearni	[Apt´e et al. 1994]		.805			
Ripper	skupni linearni	[Cohen and Singer 1999]	.683	.811		.820	
SleepingeExperts	skupni linearni	[Cohen and Singer 1999]	.753	.759		.827	
Dl-Esc	skupni linearni	[Li and Yamanishi 1999]				.820	
Charade	skupni linearni	Moulinier and Ganascia 1996]		.738			
Charade	skupni linearni	[Moulinier et al. 1996]		.783 (*F*1)			
LLSF	regresijski	[Yang 1999]		.855	.810		
LLSF	regresijski	[Yang and Liu 1999]				.849	
BalancedWinnow	on-line linearni	[Dagan et al. 1997]	.747 (M)	.833 (M)			
Widrow-Hoff	on-line linearni	[Lam and Ho 1998]				.822	
Rocchio	grupirani linearni	[Cohen and Singer 1999]	.660	.748		.776	
Findsim	grupirani linearni	[Dumais et al. 1998]				.617	.646
Rocchio	grupirani linearni	[Joachims 1998]				.799	
Rocchio	grupirani linearni	[Lam and Ho 1998]				.781	
Rocchio	grupirani linearni	[Li and Yamanishi 1999]				.625	
Classi	neur. mreže	[Ng et al. 1997]		.802			
NNET	neur. mreže	Yang and Liu 1999]				.838	
	neur. mreže	[Wiener et al. 1995]			.820		
GIS-W		[Lam and Ho 1998]				.860	
k-NN	metode učenja	[Joachims 1998]				.823	
k-NN	na temelju	[Lam and Ho 1998]				.820	
k-NN	primjera	[Yang 1999]	.690	.852	.820		
k-NN		[Yang and Liu 1999]				.856	
	SVM	[Dumais et al. 1998]				.870	.920
SVM Light	SVM	[Joachims 1998]				.864	
SVM Light	SVM	[Li Yamanishi 1999]				.841	
SVM Light	SVM	[Yang and Liu 1999]				.859	
AdaBoost.MH	kolekcija	[Schapire and Singer 2000]		.860			
	kolekcija	[Weiss et al. 1999]				.878	
	Bayes-ove	[Dumais et al. 1998]				.800	.850
	mreže	[Lam et al. 1997]	.542 (M*F*1)				

Pomoću neizravne usporedbe došlo se do sljedećih zaključaka:

- Kolekcije klasifikatora, metoda potpornih vektora – SVM, metode koje se temelje na primjerima i metode s regresijom donose vrhunske rezultate. Nema dovoljno dokaza da se donese konačan sud koja od tih metoda je najbolja.
- Neuronske mreže i on-line linearni klasifikatori rade jako dobro, iako malo lošije od prije spomenutih metoda.
- Rocchio algoritam i naivni Bayesov algoritam daju najlošije rezultate od algoritama koje se temelje na strojnom učenju.
- Podaci u tablici nisu dovoljni da bi se zaključilo nešto o stablima odluke. Međutim, u radu Dumaisa (1998) [5] klasifikator pomoću stabla odluke se pokazao tek nešto lošiji od SVM klasifikatora.

Važno je napomenuti da zaključci izneseni gore ne mogu biti apsolutni. Za drugi kontekst mogu doći do izražaja druge značajke od onih koje su došle do izražaja za Reuters, a različiti klasifikatori mogu drugačije reagirati na te značajke.

10 Literatura

[1] T. Joachims: *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*, Universitaet Dortmund, 1997.

[2] F. Sebastiani: *Machine Learning in Automated Text Categorization*, Consiglio Nazionale delle Ricerche, 2002.

[3] Y. Yang: *A Comparative Study on Feature Selection in Text Categorization*, Carnegie Melon University, 1997.

[4] N. Cristianini, J. Shawe-Taylor: *An Introduction to Support Vector Machines (and other kernel based learning methods)*, Cambridge University Press, Cambridge. 2000.

[5] S. T. Dumais, H. Chen: *Hierarchical classification of Web content,* 2000.

[6] *Machine Learning*, URL: http://www.statsoft.com/textbook/stathome.html?stmachlearn.html&1 (1.4.2008)

[7] M. E. Ruiz, P.Srinivasan: *Automatic Text Categorization Using Neural Networks*, URL: http://informatics.buffalo.edu/faculty/ruiz/publications/sigcr97/sigcrfinal2.html, (2.4.2008)

[8] C. J. van Rijsbergen: *Information Retrieval,* 1979 URL: http://www.dcs.gla.ac.uk/Keith.

[9] D. D. Lewis: *Evaluating and optmizing autonomous text classification systems*,1995

[10] R.Aheel: *Postupak klasifikacije teksta temeljen na k-nn metodi i naivnom Bayesovom klasifikatoru*, diplomski rad br. 1410, Zagreb, 2003.

[11] M.Malenica: *Primjena jezgrenih metoda u kategorizaciji teksta*, diplomski rad br. 1505, Zagreb, 2004.

[12] A.Cvitaš: *Automatsko indeksiranje dokumenata u modelu vektorskog prostora*, diplomski rad br. 1543, Zagreb, 2005.