STROJ S POTPORNIM VEKTORIMA

Sadržaj

1. Binarna klasifikacija

Mnogi binarni problemi u raspoznavanju uzoraka se rješavaju pronalaskom decizijske funkcije (hiperravnina) koja razdvaja dva skupa uzoraka. Takvih decizijskih funkcija ima beskonačno mnogo (uz pretpostavku da su razredi linearno separabilni).

slika

Dvije decizijske funkcije su prikazane na slici. Kao što se može uočiti obje pravilno razvrstavaju uzorke za učenje, no kada bi se moglo birati koju kasnije koristiti logičan odabir bi bio ona koja je označena punom linijom. Razlog tome je taj da funkcija označena punom linijom omogućuje i jednom i drugom razredu podjednake fluktuacije, tj. isprekidana linija ne omogućuje uzorcima iz razreda ω_1 nikakve pomake u desno. To svojstvo se naziva mogućnost generaliziranja i vrlo je važno svojstvo svakog klasifikatora jer ono određuje koliko će dobro klasifikator klasificirati uzorke izvan skupa za učenje. Cilj stroja s potpornim vektorima (*eng.support vector machine ,SVM*) je upravo pronaći takvu decizijsku funkciju kod koje je udaljenost do oba razreda maksimalna.

1.1 Linearno separabilni razredi

Neka je zadano N uzoraka x_i (i = 1,…,N) koji pripadaju razredu ω_1 ili ω_2. Budući da su razredi linearno separabilni decizijska funkcija je oblika

$$d(x) = w^T x + w_0 \tag{1.1}$$

uz uvjet razvrstavanja

$$w^T x_i + w_0 \begin{cases} > 0 & x_i \in \omega_1 \\ < 0 & x_i \in \omega_2 \end{cases} \tag{1.2}$$

Cilj je maksimizirati udaljenost najbližih elemenata iz razreda ω_1 i ω_2 do decizijske funkcije.
Poznato je da je udaljenost neke točke do decizijske funkcije d(x) jednaka

$$\frac{|d(x)|}{\|w\|} \tag{1.3}$$

Funkcija d(x) se može skalirati na taj način da u najbližoj točci razreda ω_1 poprimi vrijednost 1 ,a u najbližoj točci razreda ω_2 poprimi vrijednost -1.
Duljina margine nakon skaliranja iznosi

$$\frac{1}{\|w\|} + \frac{1}{\|w\|} = \frac{2}{\|w\|} \tag{1.4}$$

Sad se i uvjet razvrstavanja može zapisati u obliku:

$$w^T x_i + w_0 \begin{cases} \geq 1 & x_i \in \omega_1 \\ \leq -1 & x_i \in \omega_2 \end{cases} \tag{1.5}$$

Optimizacijski problem koji se mora riješiti je pronaći minimum funkcije (1.6) uz ograničenje (1.7).

$$J(w) = \frac{1}{2}\|w\|^2 \tag{1.6}$$

$$y_i(w^T x_i + w_0) \geq 1 \quad \textbf{(1.7)}$$

y_i je pomoćna varijabla koja ovisno o uzorku poprima vrijednost 1 ili -1 ($y_i = 1$ ako $x_i \in \omega_1$ odnosno $y_i = -1$ ako $x_i \in \omega_2$).
Ovo predstavlja kvadratni optimizacijski problem ograničen linearnom nejednadžbom.
Problem se može zapisati u neograničenom obliku.

$$F(w, w_0, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i \left[y_i (w^T x_i + w_0) - 1 \right] \quad \textbf{(1.8)}$$

Ovaj oblik se još naziva i Lagrangeova funkcija, a α_i Lagrangeovi multiplikatori.
Optimum zadovoljava KKT(Karush-Kuhn-Tucker) uvjete dane jednadžbama (1.9) - (1.12).

$$\frac{\partial}{\partial w} F(w, w_0, \alpha) = 0 \quad \textbf{(1.9)}$$

$$\frac{\partial}{\partial w_0} F(w, w_0, \alpha) = 0 \quad \textbf{(1.10)}$$

$$\alpha_i \geq 0 \quad \textbf{(1.11)}$$

$$\alpha_i \left[y_i (w^T x_i + w_0) - 1 \right] = 0 \quad \textbf{(1.12)}$$

Uvrštavanjem (1.8) u (1.9) i (1.10) dobivaju se jednadžbe (1.13) i (1.14).

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \quad \textbf{(1.13)}$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \quad \textbf{(1.14)}$$

Iz jednadžbe (1.12) se može uočiti da je umnožak Lagrangeovog multiplikatora i iznosa decizisjke funkcije umanjenog za 1 pripadne točke uvijek jednak 0. Točke koje su udaljene za 1 od decizijske funkcije (te je time njihov iznos umanjen za 1 jednak 0) su upravo točke koje se nalaze na rubovima margina. U tim slučajevima pripadni Lagrangeov multiplikator može biti različit od 0. Te točke se nazivaju potporni vektori. Tu valja obratiti pozornost da nije nužno da sve točke koje se nalaze na rubu margina budu ujedno i potpornji vektori. Pripadni Lagrangeov multiplikator točaka koje se ne nalaze na rubu margine tj. za njih vrijedi $d(x)>1$ iznosi 0. To se može protumačiti na način da te točke nisu važne za određivanje decizijske funkcije i njihovim uklanjanjem se ne utječe na samu decizijsku funkciju.
Uvrštavanjem (1.13) i (1.14) u (1.8) dobiva se sljedeći optimizacijski problem uz uvjet (1.14).

$$\max_{\alpha} f(\alpha) = \left(\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j \right) \quad \textbf{(1.15)}$$

Ovaj oblik se još nazva i dualni oblik.

Normala ravnine je zadana jednadžbom (1.13) dok je slobodna konstanta (w_0) dana izrazom.

$$w_0 = y_i - w^T x_i \tag{1.16}$$

Sa gledišta numeričke preciznosti bolje je uzeti izraz (1.17)

$$w_0 = \frac{1}{S}\sum_{i \in S}\left(y_i - w^T x_i\right) \tag{1.17}$$

Sada se decizijska funkcija može zapisati u obliku

$$d(x) = \sum_{i \in S} \alpha_i y_i x_i^T x + w_0 \tag{1.18}$$

sa uvjetom razvrstavanja

$$d(x_i)\begin{cases} > 0 & x_i \in \omega_1 \\ < 0 & x_i \in \omega_2 \end{cases} \tag{1.19}$$

1.2 Linearno neseparabilni razredi

Najčešće su razredi linearno neseparabili i model predstavljen u prethodnom poglavlju ne može pronaći decizijsku funkciju.
Da bi se prilagodio postojeći model SVM-a linearno neseparabilnim razredima, uvodi se nova varijabla – nenegativna varijabla ξ_i.
Uvjet zadan jednadžbom (1.7) postaje uvjet zadan jednadžbom (1.20)

$$y_i(w^T x_i + w_0) \geq 1 - \xi_i \tag{1.20}$$

Uvođenjem varijable ξ_i postoji prihvatljiva decizijska funkcija. Ovisno o varijabli ξ_i postoje tri kategorije uzoraka:

1) ξ_i=0 uzorci koji pripadaju linearno separabilnom dijelu i obrađeni su u prethodnom poglavlju
2) ξ_i >0 i $\xi_i \leq 1$ uzorci koji više nemaju maksimalnu marginu, ali još uvijek su točno klasificirani
3) ξ_i>1 uzorci koji su pogrešno klasificirani

Sada je potrebno pronaći takvu decizijsku funkciju u kojoj je broj uzoraka koji nemaju maksimalnu marginu minimalan.
Potrebno je pronaći minimum funkcije

$$J(w) = \sum_{i=1}^{M} I(\xi_i) \tag{1.21}$$

gdje je

$$I(\xi_i) = \begin{cases} 1 & \xi_i > 0 \\ 0 & \xi_i = 0 \end{cases} \tag{1.22}$$

Ako se uzme u obzir i maksimizacija margine dobiva se sljedeća funkcija koju je potrebno minimizirati.

$$J(w, w_0, \xi) = \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{N} I(\xi_i) \tag{1.23}$$

uz uvjete

$$y_i (w^T x_i + w_0) \geq 1 - \xi_i \tag{1.24}$$

$$\xi_i \geq 0 \tag{1.25}$$

C je parametar koji određuje važnost širine margine i broja uzoraka koji nemaju maksimalnu marginu.
Lagrangeov oblik danog problema dan je funkcijom

$$F(w, w_0, \xi, \alpha, \mu) = \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{N} I(\xi_i) - \sum_{i=1}^{N} \mu_i \xi_i - \sum_{i=1}^{N} \alpha_i \left[y_i \left(w^T x_i + w_0 \right) - 1 + \xi_i \right] \tag{1.26}$$

KKT uvjeti su

$$\frac{\partial}{\partial w} F(w, w_0, \xi, \alpha, \mu) = 0 \tag{1.27}$$

$$\frac{\partial}{\partial w_0} F(w, w_0, \xi, \alpha, \mu) = 0 \tag{1.28}$$

$$\frac{\partial}{\partial \xi} F(w, w_0, \xi, \alpha, \mu) = 0 \tag{1.29}$$

$$\alpha_i \geq 0, \mu_i \geq 0 \tag{1.30}$$

$$\alpha_i \left[y_i (w^T x_i + w_0) - 1 + \xi_i \right] = 0 \tag{1.31}$$

$$\mu_i \xi_i = 0 \tag{1.32}$$

Iz (1.26) i (1.29) slijedi

$$C - \mu_i - \alpha_i = 0 \tag{1.33}$$

Dualni oblik je

$$\max_{\alpha} f(\alpha) = \left(\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j \right) \tag{1.34}$$

uz uvjete

$$0 \leq \alpha_i \leq C \tag{1.35}$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{1.36}$$

2. Klasifikacija više razreda uzoraka

Iako možda SVM nije najpogodnija metoda za rješavanje problema raspoznavanja velikog broja razreda uzoraka(npr. 200 razreda) razvijene su mnoge metode koje daju zadovoljavajuće rezultate. Najveće probleme naravno, predstavlja samo učenje koje se zasniva na kvadratnom programiranju. Veliki broj razreda i veliki broj uzoraka za učenje može pretvoriti kvadratno programiranje u vrlo neefikasnu metodu koja zahtijeva veliku količinu memorije, a samo učenje traje vrlo dugo. Drugi problem koji se javlja kod klasifikacije više razreda su mogućnost pojavljivanja nedefiniranih regija. Za oba problema postoje efikasni pristupi koji ih rješavaju.
Neke od metoda su :

1. jedan nasuprot svih metoda
2. usporedba po parovima
3. stabla odluke

Ideja jedan nasuprot svih metode je odvajanje decizijskom funkcijom jednog razreda od svih preostalih. Pritom je potrebno izgraditi k decizijskih funkcija, gdje je k broj razreda.
Neka je formulom (2.1) zadana decizijska funkcija $d_i(x)$.

$$d_i(x) = w_i^T x + w_0 \qquad \textbf{(2.1)}$$

Tada će uzorak biti razvrstan u razred ω_i ako je zadovoljen uvjet (2.2)

$$d_i(x) > 0 \qquad \textbf{(2.2)}$$

Može se odmah uočiti jedan vrlo važan nedostatak ove metode, a to je pojava nedefiniranih regija. Tako klasifikator neće znati klasificirati slučaj kada je $d_1(x)>0$, $d_2(x)>0$, $d_3(x)<0$ ili slučaj kada je $d_1(x)<0$, $d_2(x)<0$, $d_3(x)<0$. Ovo je takozvana diskretna verzija decizijskih funkcija jer se uzima samo podatak je li iznos funkcije u danoj točci veći ili manji od 0.
Da bi se izbjeglo pojavljivanje nedefiniranih regija, uvodi se kontinuirani oblik decizijske funkcije dan izrazom (2.3).

$$\arg\max_{i=1,\ldots,k} d_i(x) \qquad \textbf{(2.3)}$$

Sada će uzorak biti razvrstan u razred čija pripadna decizijska funkcija ima najveću vrijednost. Ovo je samo jedan od načina kojim se može riješiti problem nedefiniranih područja.

Usporedba po parovima se temelji na izračunavanju $\frac{k(k-1)}{2}$ decizijskih funkcija, gdje je k broj razreda. Prilikom klasifikacije potrebno je izračunati vrijednosti $\frac{k(k-1)}{2}$ decizijskih funkcija i odlučiti se na temelju rezultata kojem razredu pripada nepoznati uzorak. Samo učenje decizijskih funkcija kao i kasnije testiranje i klasifikacija mogu biti neefikasni upravo zbog vrlo velikog broja samih decizijskih funkcija.

2.1 Klasifikacija temeljena na stablima odluke

Stabla odluke su jedna od metoda kojom se može riješiti problem nedefiniranih područja koja se pojavljuju kod metode jedan nasuprot svih. Ideja je da se u svakom čvoru stabla provodi binarna klasifikacija, tj. da se odvoji jedna grupa razreda od druge. Potrebno je dati algoritam koji će po nekom pravilu grupirati razrede u posebne grupe. Algoritam

grupiranja je posebice važan za ovaj postupak jer on ima vrlo veliki utjecaj na konačnu točnost samog klasifikatora.
Do sada je razvijeno više različitih algoritama grupiranja , a osnovna razlika im je rade li grupiranje u neproširenom prostoru značajki ili proširenom prostoru značajki koji nastaje nakon proširenja pomoću jezgrenih funkcija (*eng. kernel space*).
Ovdje će biti predstavljen samo jedan algoritam grupiranja i to algoritam koji radi u neproširenom prostoru značajki.
Cilj grupiranja je dobivanje dvije grupe razreda u pojedinom čvoru stabla. Ako se svi uzorci iz jedne grupe (dobivene grupiranjem) označe s y=+1, a iz druge grupe s y=-1 dobiva se problem binarne klasifikacije koja se može riješiti s SVM-om. Da bi se odredilo koji razred pripada kojoj grupi potrebno je imati mjeru udaljenosti. Također je potrebno za svaki razred imati predstavnika. Taj predstavnik jednostavno može biti centar razreda koji se jednostavno dobiva formulom (2.4).

$$c_i = \frac{1}{|X_i|} \sum_{x \in X_i} x \qquad \textbf{(2.4)}$$

Sada se udaljenost između dva razreda može jednostavno dobiti kao Euklidska udaljenost između centara razreda što je i dano u formuli (2.5).

$$l_{ij} = \| c_i - c_j \| \qquad \textbf{(2.5)}$$

1) $grupa_1 = \{\emptyset\}$;
2) $grupa_2 = \{\emptyset\}$;
3) izračunaj c_i svih razreda
4) {r1,r2} = $\max_{i \neq j} l_{ij}$;
5) $grupa_1 = grupa_1 \cup r_1$;
6) $grupa_2 = grupa_2 \cup r2$;
7) **ponavljaj dok** nisu svi razredi u jednoj od grupa
8) r = $\min_{i \neq grupa1} l_{igrupa1}$;
9) p = $\min_{i \neq grupa2} l_{igrupa2}$;
10) **ako je** $l_{r,grupa1} < l_{p,grupa2}$
11) $grupa_1 = grupa_1 \cup r$;
12) **inače**
13) $grupa_2 = grupa_2 \cup p$;
14) **kraj**
15) **vrati** {$grupa_1$,$grupa_2$}

Slika 2.1 Pseudokod algoritma grupiranja

U koracima 1) do 3) radi se inicijalizacija algoritma. U koracima 4) do 6) traže se dva najudaljenija razreda koji se dodjeljuju svaki u po jednu grupu. U koracima 7) do 13) traži

se razred koji je najbliži pojedinoj grupi te se taj razred i toj grupi dodjeljuje. Ti koraci se ponavljaju sve dok se svi razredi ne dodijele u jednu od dvije grupe.
Izgradnja stabla je vrlo jednostavna. Nakon što se provede grupiranje dobivaju se dvije grupe. Pronalazak decizijske funkcije za te dvije grupe se temelji na binarnom SVM-u. Nakon toga se nad dobivenim razredima iz grupa (dobivenih grupiranja) rekurzivno ponavlja postupak i tako sve dok u jednom čvoru ne preostane samo jedan razred. Taj čvor je list i on nosi oznaku tog razreda.
Klasifikacija uzorka se provodi u svakom čvoru stabla koji se nalazi na putu do određenog lista. Klasifikacija u pojedinom čvoru se koristi za usmjeravanje u lijevo ili desno podstablo ovisno o rezultatu klasifikacije.
Postupak rješavanja problema klasifikacije više razreda koji se temelji na binarnim stablima odluke ima nekoliko dobrih svojstava.

I. potrebno je pronaći svega k-1 decizijskih funkcija
II. samo se u korijenu razmatraju svi uzorci za treniranje dok je u ostalim čvorovima taj skup sve manji i manji. To znatno pridonosi samo brzini treniranja.
III. tijekom klasifikacije se ispituju samo neke decizijske funkcije

.

3. Treniranje SVM-a

U prethodnim poglavljima je pokazano da se rješenje nalazi rješavanjem problema koji je zadan u obliku kvadratnog programiranja. Postoje numerički postupci koji rješavaju probleme kvadratnog programiranja, no oni se pokazuju kao neučinkoviti za SVM. Dva su razloga za to: 1) Q matrica može biti jako velika 2) samo rješavanje problema može trajati jako dugo. Kao što je poznato do nedavno (prije 10 godina) radna memorija je bila veliki luksuz i gotovo niti jedno kućno računalo nije imalo preko 100 MB. Ako se uzme u obzir da problem od 4000 uzoraka za učenje (koji se inače smatra malim problemom) može imati 128 MB veliku matricu očito je bilo potrebno smisliti nove načine rješavanja problema.
Radi jednostavnijeg pisanja u sljedećim poglavljima sve će biti pisano u obliku matrica i vektora.

Dualni problem se sada može zapisati u sjedećem obliku

$$\min_{\alpha} f(\alpha) = \frac{1}{2}\alpha^T Q\alpha - e^T\alpha \tag{3.1}$$

uz uvjete

$$\begin{aligned} &0 \le \alpha_i \le C, i = 1,...n \\ &y^T\alpha = 0 \end{aligned} \tag{3.2}$$

Dodatno još vrijedi da je vektor *e* vektor koji sadrži sve jedinice. Matrica Q je n×n matrica. Također vrijedi da je $Q_{ij}=y_iy_jK(x_i,x_j)$. $K(x_i,x_j)$ je jezgrena funkcija.

3.1 Dekompozicija

Ideja dekompozicije jest dijeljenje skupa Lagrangeovih multiplikatora(W) u dva skupa B i N. Pritom mora vrijediti $B \bigcup N = W$ i $B \bigcap N = 0$ Na skup B se tada može primijeniti neka od metoda za rješavanje kvadratnog problema , a skup N se drži konstantnim.
sada se jednadžba 2.6 može prepisati u sljedeći oblik

$$\begin{aligned} &\min_{\alpha} f(\alpha) = \frac{1}{2}\begin{bmatrix}\alpha_B^T, \alpha_N^T\end{bmatrix}\begin{bmatrix}Q_{BB} & Q_{BN} \\ Q_{NB} & Q_{NN}\end{bmatrix}\begin{bmatrix}\alpha_B \\ \alpha_N\end{bmatrix} - \begin{bmatrix}e_B^T & e_N^T\end{bmatrix}\begin{bmatrix}\alpha_B \\ \alpha_N\end{bmatrix} = \\ &= \frac{1}{2}\alpha_B^T Q_{BB}\alpha_B + \left(-e_B + Q_{BN}\alpha_N\right)^T\alpha_B + kons\tan ta \end{aligned} \tag{3.3}$$

uz uvjete

$$\begin{aligned} &0 \le \alpha_B \le C, i = 1,...n \\ &y_B^T\alpha = -y_N^T\alpha \end{aligned} \tag{3.4}$$

Iz drugog uvjeta se uočava jedno vrlo važno svojstvo dekompozicije , a to je $|B| \ge 2$. Jasno je kada bi se samo jedan Lagrangeov multiplikator mijenjao prilikom promjene odmah bi se prekršilo pravilo iz jednadžbe 2.7 , no ako se mijenja dva ili više multiplikatora tada se uvijek mogu namjestiti tako da jednadžba 2.7 ostane zadovoljena.
Budući da optimalno rješenje mora zadovoljavati KKT uvjete nakon optimiranja jednog podskupa Lagrangeovih multiplikatora odabire se novi i tako sve dok nisu zadovoljeni KKT uvjeti.

Ovisno o skupu B postoje dvije vrste metoda dekompozicije.

- metoda koja u svakom koraku ima konstantu veličinu skupa B
- metoda koja ima varijabilnu veličinu skupa B

```
1) dok je prekršeno pravilo optimalnosti
2)       odaberi |B| varijabli za skup B;
3)       preostale varijable su fiksirane;
4)       riješi QP - podproblem nad B ;
5) kraj
6) vrati α;
```

Slika 3.1 Pseudo kod algoritma dekompozicije

3.2 SMO (Sequential Minimal Optimization)

SMO metoda je metoda dekompozicije koja ima stalnu veličinu skupa B i to $|B| = 2$.

Sada se jednadžba 2.8 može napisati kao

$$\min_{\alpha} f(\alpha) = \frac{1}{2}\begin{bmatrix}\alpha_i, \alpha_j\end{bmatrix}\begin{bmatrix}Q_{ii} & Q_{ij} \\ Q_{ij} & Q_{jj}\end{bmatrix}\begin{bmatrix}\alpha_i \\ \alpha_j\end{bmatrix} - \left(-e_B + Q_{BN}\alpha_N\right)^T\begin{bmatrix}\alpha_i \\ \alpha_j\end{bmatrix} + kons\tan ta \qquad \textbf{(3.5)}$$

uz uvjete

$$\begin{aligned} &0 \le \alpha_i, \alpha_j \le C, i = 1,...n \\ &y_i\alpha_i + y_j\alpha_j = -y_N^T\alpha \end{aligned} \qquad \textbf{(3.6)}$$

Budući da se optimizacija u jednom koraku provodi samo nad dva Lagrangeova multiplikatora rješenje se dobiva analitički. Takav postupak je vrlo brz što je pozitivno za algoritam, jer više nije potrebno gubiti puno vremena na optimizatoru koji koristi neki numerički postupak.

Odabir samih multiplikatora u jednom koraku nije toliko jednostavan. Do sada je razvijeno više postupaka.

Keerthi u [3] predlaže korištenje sljedećeg postupka.

Neka su definirana dva skupa I_{up} i I_{low} na sljedeći način.

$$\begin{aligned} I_{up}(\alpha) &\equiv \{t \mid \alpha_t < C, y_t = 1 \forall \alpha_t > 0, y_t = -1\} \\ I_{low}(\alpha) &\equiv \{t \mid \alpha_t < C, y_t = -1 \forall \alpha_t > 0, y_t = 1\} \end{aligned} \qquad \textbf{(3.7)}$$

Tada se par (B={i,j}) za optimizaciju može odabrati na sljedeći način.

$$\begin{aligned} i &\in \arg\max_t \{-y_t \nabla f(\alpha)_t \mid t \in I_{up}(\alpha)\} \\ j &\in \arg\min_t \{-y_t \nabla f(\alpha)_t \mid t \in I_{low}(\alpha)\} \end{aligned} \qquad \textbf{(3.8)}$$

gdje je

$$\nabla f(\alpha) \equiv Q\alpha - e \qquad \textbf{(3.9)}$$

$\nabla f(\alpha)$ je gradijent kriterijske funkcije.

Ovaj postupak se može izvesti iz KKT uvjeta jer je poznato da optimalno rješenje mora zadovoljavati KKT uvjete. Izvod se može pronaći u [3]. Postupak u svakom koraku pronalazi par koji najviše krši KKT uvjete te nad njima provodi optimizaciju.
Joachims([5]) predlaže korištenje aproksimacije prvog reda kriterijske funkcije za određivanje skupa varijabli (u slučaju SMO samo dvije varijable) čije će računanje dovesti do najvećeg napretka.
Neka je aproksimacija prvog reda kriterijske funkcije zadana sljedećom formulom

$$f(\alpha + d) \approx f(\alpha) + \nabla f(\alpha) d = f(\alpha) + \nabla f(\alpha)_B^T d_B \tag{3.10}$$

Vektor d označava smjer najbržeg spusta. Da bi se dobio vektor d potrebno je riješiti sljedeći optimizacijski problem.

$$\begin{aligned} &\min_{d_B} \nabla f(\alpha)_B^T d_B \\ &uvjeti: \\ &y_B^T d_B = 0 \\ &d_t \geq 0 \quad ako \quad \alpha_t = 0, t \in B \\ &d_t \leq 0 \quad ako \quad \alpha_t = C, t \in B \\ &-1 \leq d_t \leq 1, t \in B \end{aligned} \tag{3.11}$$

Može se pokazati da je postupak koji traži par koji najviše krši KKT uvjete povezan s postupkom koji se temelji na aproksimaciji prvog reda kriterijske funkcije.

Budući da je kriterijska funkcija kvadratna logično je koristiti aproksimaciju drugog reda.

$$\begin{aligned} &\Delta f(\alpha) = f(\alpha + d) - f(\alpha) = \nabla f(\alpha)^T d + \frac{1}{2} d^T \nabla^2 f(\alpha) d = \\ &\nabla f(\alpha)_B^T d_B + \frac{1}{2} d_B^T \nabla^2 f(\alpha)_{BB} d_B \end{aligned} \tag{3.12}$$

Da bi se sada našao smjer d potrebno je rješiti sljedeći optimizacijski problem.

$$\begin{aligned} &\min_{d_B} \frac{1}{2} d_B^T \nabla^2 f(\alpha)_{BB} d_B + \nabla f(\alpha)_B^T d_B \\ &uvjeti: \\ &y_B^T d_B = 0 \\ &d_t \geq 0 \quad ako \quad \alpha_t = 0, t \in B \\ &d_t \leq 0 \quad ako \quad \alpha_t = C, t \in B \end{aligned} \tag{3.13}$$

Rješavanje ovog problema nije jednostavno i potrebno je ispitati sve moguće parove.
Ispitivanje svih mogućih kombinacija se može zaobići na sljedeći način. Multiplikator *i* se odabire na način predložen u (3.8). Za multiplikator *j* se uvodi modifikacija.
Da bi se došlo de te modifikcije potrebno je proučiti (3.13).
Definiraju se varijable :

$$\begin{aligned} &\hat{d}_i = y_i d_i \\ &\hat{d}_j = y_j d_j \\ &\hat{d}_i = -\hat{d}_j \; iz \quad y_B^T d_B = 0 \end{aligned} \tag{3.14}$$

Raspisivanjem (3.13) dobiva se sljedeći izraz:

$$\frac{1}{2}\begin{bmatrix} d_i , d_j \end{bmatrix}\begin{bmatrix} Q_{ii} & Q_{ij} \\ Q_{ij} & Q_{jj} \end{bmatrix}\begin{bmatrix} d_i \\ d_j \end{bmatrix}+\begin{bmatrix} \nabla f(\alpha)_i & \nabla f(\alpha)_j \end{bmatrix}\begin{bmatrix} d_i \\ d_j \end{bmatrix}= \\ \frac{1}{2}\left(K_{ii}+K_{jj}-2K_{ij}\right)\hat{d}_j^2+\left(-y_i\nabla f(\alpha)_i+y_j\nabla f(\alpha)_j\right)\hat{d}_j \tag{3.15}$$

Kako je $K_{ii}+K_{jj}-2K_{ij}>0$ definiraju se varijable

$$a_{ij}=K_{ii}+K_{jj}-2K_{ij}>0 \\ b_{ij}=-y_i\nabla f(\alpha)_i+y_j\nabla f(\alpha)_j>0 \tag{3.16}$$

(3.15) ima minimum u

$$\hat{d}_j=-\hat{d}_i=-\frac{b_{ij}}{a_{ij}}<0 \tag{3.17}$$

a iznos (3.13) u točki minimuma jest

$$-\frac{b_{ij}^2}{2a_{ij}}$$

Ako je K pozitivno definitna matrica tada je za svaki $i\neq j$ vrijedi $K_{ii}+K_{jj}-2K_{ij}>0$.
Sada se može napisati novi kriterij za odabir para {i,j}.

$$i\in\arg\max_t\left\{-y_t\nabla f(\alpha)_t \mid t\in I_{up}(\alpha)\right\} \\ j\in\arg\min_t\left\{-\frac{b_{it}^2}{a_{it}} \mid t\in I_{low}(\alpha),-y_t\nabla f(\alpha)_t<-y_i\nabla f(\alpha)_i\right\} \tag{3.18}$$

Ako je $K_{ii}+K_{jj}-2K_{ij}\leq 0$ tada se kriterij (3.18) ne može primijeniti za izbor para {i,j}. Pokazano je u [4] da se taj problem rješava uvođenjem dodatnog izraza u kriterijsku funkciju što na kraju rezultira rezultatom $a_{ij}\equiv\tau$, gdje je τ proizvoljno mali broj.

Sada se konačno može dati postupak izbora para Lagrangeovih multiplikatora u jednoj iteraciji optimizacije.
1. Definiraju se a_{ts} i b_{ts} kao u (3.16).

$$\bar{a}_{ts}\equiv\begin{cases} a_{ts} & ako \quad a_{ts}>0 \\ \tau & ako \quad a_{ts}\leq 0 \end{cases} \tag{3.19}$$

2. Odabir {i,j}

$$i\in\arg\max_t\left\{-y_t\nabla f(\alpha)_t \mid t\in I_{up}(\alpha)\right\} \\ j\in\arg\min_t\left\{-\frac{b_{it}^2}{\bar{a}_{it}} \mid t\in I_{low}(\alpha),-y_t\nabla f(\alpha)_t<-y_i\nabla f(\alpha)_i\right\} \tag{3.20}$$

3. Vraćanje B ={i,j}

3.3 Dodatna poboljšanja

Dodatna poboljšanja gore navedenog postupka se mogu postići na dva područja. To su korištena memorija i korišteno vrijeme.

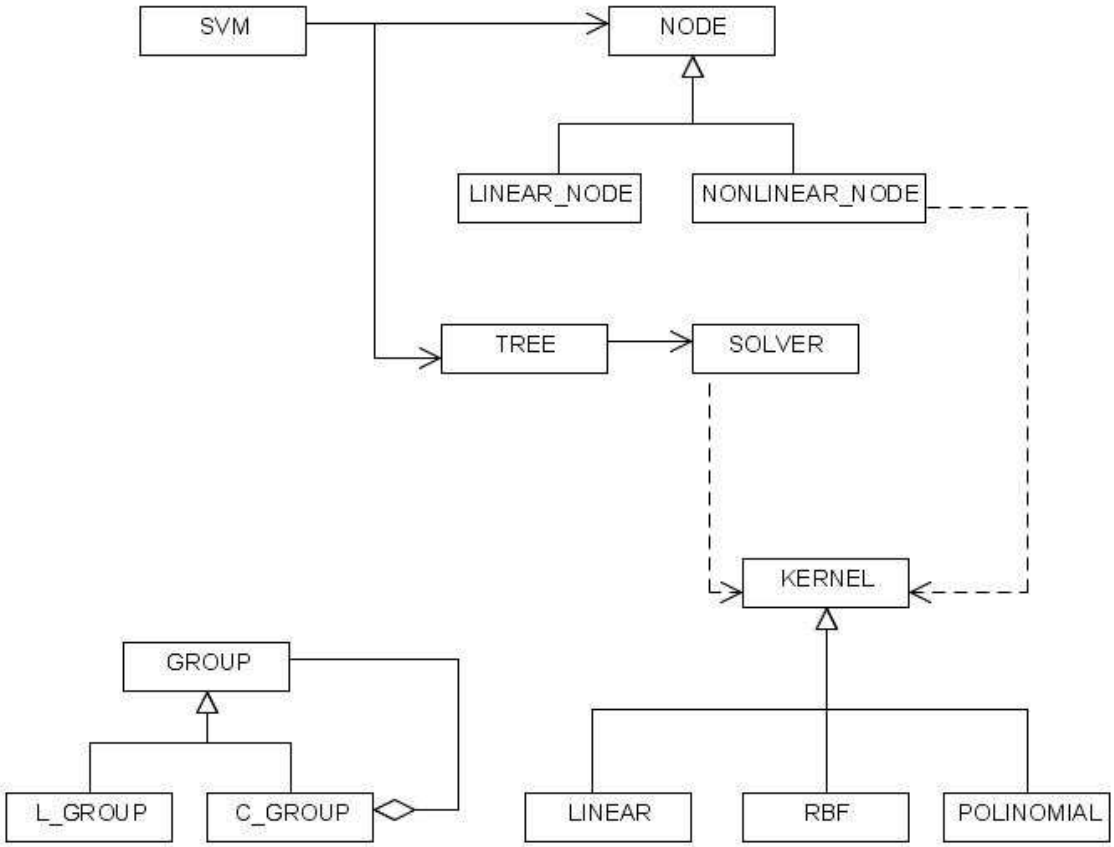
Ako za treniranje nije dostupna velika količina radne memorije (to nekada može biti i samo nekoliko MB) tada opcija spremanja cijele jezgrene matrice (Q) u memoriju ne dolazi u obzir. Ponovno računanje jezgrene funkcije za par uzoraka prilikom svakog korištenja moglo bi znatno usporiti sami algoritam. Uz pretpostavku da je ipak dostupna određena količina radne memorije (možda samo 100KB) može se ipak dobiti dovoljno brzi algoritam. Ideja je da se u memoriji čuvaju oni stupci (ili redci ,budući da je matrica simetrična svejedno je) matrice koji su korišteni najčešće u posljednjih nekoliko koraka. Kada se dogodi promašaj tada se briše onaj stupac koji je najdavnije korišten , a dodaje se novi. Računa se cijeli stupac , a ne samo jedan element jer kad je potreban jedan element najčešće je potreban čitavi stupac u kojem se taj element nalazi.

Kako bi se dodatno ubrzao sami algoritam koristi se metoda smanjivanja problema (*eng. shrinking*). Ideja metode se zasniva na činjenici da su za decizijsku funkciju važni samo potporni vektori. Ako se uklone preostali uzorci na sami rezultat se neće utjecati. Također broj potpornih vektora je redovito puno manji od samog broja uzoraka pa je i za očekivati da će ubrzanje biti znatno.

Nažalost ove metode još nisu implementirane u samoj implementaciji klasifikatora. Implementirani klasifikator sprema cijelu Q matricu u memoriju. To mu omogućuje da računanje jezgrene funkcije za jedan par uzoraka obavi samo jedanput što pridonosi brzini. S druge strane postupak ovisno o broju uzoraka za treniranje može zauzeti i nekoliko desetaka MB radne memorije. Ako je problem veliki (nekoliko desetaka tisuća uzoraka za učenje) može se dogoditi da neće biti dovoljno radne memorije.

Tijekom računanja Lagrangeovih multiplikatora klasifikator uzima u obzir sve uzorke te taj način donekle usporava sami postupak treniranja. Opet za velike probleme klasifikator bi se mogao pokazati kao neučinkovit.

4. Programsko ostvarenje SVM-a



Slika 4.1 Dijagram razreda

Dijagram razreda je prikazan na slici 4.1. Glavni razred preko kojeg se komunicira s cijelim klasifikatorom je razred *svm*. Taj razred podržava četiri operacije koje su bitne za sami klasifikator a to su : treniranje (train), klasifikacija nepoznatog uzorka (classify) spremanje (save) i učitavanje (load) svm-a iz datoteke. Razred *svm* koristi razred *tree* za izgradnju samog stabla. Iako se na slici ne može uočiti razred *tree* koristi razred *algorithm* koji definira sami postupak grupiranja. Razred *tree* koristi i razred *solver* u kojemu je ostvaren SMO postupak. Ovdje je važno napomenuti da prilikom izgradnje samog stabla traženje Lagrangeovih multiplikatora za svaki čvor pomoću *solver* razreda je neovisno. To omogućuje uvođenje paralelizacije samog postupka čime bi samo treniranje bilo još brže.
Na slici se također uočavaju tri vrlo važne komponente za sami sustav. To su *group*, *kernel* i *node*.
Skup uzoraka koji pripadaju istom razredu se smještaju u razred *l_group*. Taj razred još sadrži neke vrlo važne informacije za sami skup uzoraka kao što je njihov centar. Razred *c_group* omogućuje bilo kakvo grupiranje razreda (koji su predstavljeni razredom *l_group*) i preostalih grupa razreda (koje su predstavljene razredom *c_group*). Također *c_group* se ponaša kao običan razred pa je moguće saznati i neke važne informacije kao što su centar grupe.
Razred *kernel* omogućuje u klasifikatoru korištenje različitih jezgrenih funkcija na vrlo jednostavan način.

Razred *node* predstavlja jedan čvor stabla. Potrebne su dvije vrste čvora ovisno koja se jezgrena funkcija koristi. Razred *linear_node* koristi se prilikom korištenja linearne jezgrene funkcije dok se razred *nonlinear_node* koristi za preostale jezgrene funkcije. Razlika između ove dvije vrste čvorova je u tome što se kod korištenja linearne jezgrene funkcije može eksplicitno izračunati decizijska funkcija dok kod preostalih jezgrenih funkcija se mora pamtiti potporni vektori i pripadajući Lagrangeovi multiplikatori. Ovdje se treba napomenuti da (iako nije naznačeno na slici) stvaranjem čvorova u stablu upravljaju same jezgrene funkcije tako da je korisnik oslobođen tog posla.
Ovakva arhitektura omogućuje vrlo lagano proširenje jer su glavne komponente nezavisne tj. sva komunikacija se odvija preko apstraktnih razreda.

5. Literatura

[1] Shigeo Abe, Support Vector Machines for Pattern Classification, Springer-Verlag London 2005
[2] John C. Platt, Fast Training of Support Vector Machines using Sequential Minimal Optimization,
[3] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, K. R. K. Murthy, Improvements to Platt's SMO Algorithm for SVM Classifier Design ,u Bernhard Scholkopf, Cristopher J. C. Burges, Sebastian Mika, Advances in Kernel Methods, MIT Press 1998
[4] Rong-En Fan, Pai-Hsuen Chen, Chih-Jen Lin, Working Set Selection Using Second Order Information for Training Support Vector Machines, Journal of Machine Learning Research 11/2005.
[5] Thorsten Joachims, Making Large-Scale SVM Learning Practical, u Bernhard Scholkopf, Cristopher J. C. Burges, Sebastian Mika, Advances in Kernel Methods, MIT Press 1998