

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 852

**RASPOZNAVANJE PROMETNIH
ZNAKOVA METODOM POTPORNIH
VEKTORA**

Ivan Kusalić

Zagreb, lipanj 2009.

Sadržaj

Uvod.....	1
1. Stroj s potpornim vektorima	2
1.1. Dvije intuicije o klasifikaciji.....	2
1.1.1. Intuicija o pouzdanosti klasifikacije	2
1.1.2. Intuicija o najboljoj granici odluke	3
1.2. Klasifikator optimalne granice.....	4
1.2.1. Korištena notacija	4
1.2.2. Funkcijska i geometrijska margina	4
1.2.3. Formulacija optimizacijskog problema.....	6
1.2.4. Primal i Dual optimizacijski problem.....	8
1.3. Jezgreni trik.....	13
1.3.1. Motivacija	13
1.3.2. Ideja	13
1.3.3. Izbor jezgre	14
1.3.4. Primjeri jezgara.....	15
1.4. Stroj s potpornim vektorima L1 meke granice	17
1.4.1. Motivacija	17
1.4.2. Modifikacija.....	19
2. Slijedna minimalna optimizacija	20
2.1. Koordinatni uspon.....	20
2.2. Primjena na Stroj s potpornim vektorima	21
3. Problem raspoznavanja prometnih znakova	25
3.1. Reprezentacija prometnih znakova.....	25
3.1.1. Početne fotografije prometnih znakova	25
3.1.2. Podatci uz fotografije prometnih znakova	26
3.1.3. Normalizirane fotografije prometnih znakova.....	27
3.1.4. Izbor vektora značajki.....	27
3.2. Programsko ostvarenje Stroja s potpornim vektorima.....	28
3.2.1. Programsko ostvarenje Slijedne minimalne optimizacije.....	28

3.2.2.	Vizualizator Stroja s potpornim vektorima.....	29
3.2.3.	Sustav za prepoznavanje prometnih znakova	32
3.3.	Rezultati	33
3.4.	Moguća poboljšanja	34
Zaključak		36
Literatura.....		38
Naslov, sažetak i ključne riječi		39
Title, abstract and keywords		40

Uvod

Raspoznavanje ili klasifikacija objekta je proces određivanja pripadnosti objekta određenoj grupi objekata sa sličnim odabranim svojstvima. Kako će neki objekt biti klasificiran uvelike ovisi o odabranim svojstvima koja su izabrana kao reprezentativna za objekt. Osim o odabranim svojstvima, konačni rezultat klasifikacije ovisi i o algoritmu kojim se klasifikacija vrši.

Postoji veoma velik broj različitih algoritama za klasifikaciju, a proučavani Stroj s potpornim vektorima je jedan od njih. Stroj s potpornim vektorima pripada skupini algoritama za nadgledano strojno učenje što znači da uči funkciju na temelju danog mu skupa za učenje. Skup za učenje se sastoji od unaprijed pripremljenih ispravnih klasifikacija objekata koje su predstavljene kao parovi vektora svojstava (značajki) nekog objekta i pripadne (ispravne) klasifikacije tog objekta.

Stroj s potpornim vektorima je algoritam koji mnogi stručnjaci smatraju najučinkovitijim standardnim (eng. *of-the-shelf*) algoritmom za nadgledano strojno učenje. Iako taj stav ne dijele svi stručnjaci u području strojnog učenja, nitko ne dovodi u pitanje kvalitetu i značaj ovog algoritma.

Cilj ovog rada je smjestiti Stroj s potpornim vektorima unutar područja strojnog učenja, postaviti teorijsku osnovu algoritma, te ga primijeniti na problem klasifikacije prometnih znakova.

1. Stroj s potpornim vektorima

Stroj s potpornim vektorima (eng. *Support Vector Machine*, [1]) će biti objašnjen postepeno. Prvo će biti iznesene dvije općenite intuicije o klasifikaciji, nakon toga sljedi objašnjenje Klasifikatora optimalne granice (eng. *Optimal Margin Classifier*, [2]), te će konačno Klasifikator optimalne granice biti modificiran tako da nastane Stroj s potpornim vektorima.

1.1. Dvije intuicije o klasifikaciji

Slijede dvije općenite intuicije o klasifikaciji koje su motivatori za izgradnju proučavanih algoritama.

1.1.1. Intuicija o pouzdanosti klasifikacije

Promatra se slučaj binarne klasifikacije (klasifikacija objekata u dvije grupe). Neka jedna grupa ima oznaku „1“, a druga „-1“.

Neka je $f(x): \mathbf{R}^n \rightarrow \mathbf{R}$ funkcija odluke o klasifikaciji koja prima vektor značajki (skup konkretnih vrijednosti svojstava objekta), a vraća rezultat klasifikacije (realan broj): $y = f(x)$. Ako vrijedi $y \geq 0$, tada objekt pripada grupi s oznakom „1“, a ako vrijedi $y < 0$, tada objekt pripada grupi s oznakom „-1“.

Ako za rezultat klasifikacije vrijedi $y \gg 0$, tada je vjerojatnost da objekt zbilja pripada grupi s oznakom „1“ veoma velika, ili drugačije rečeno: odluka o klasifikaciji u grupu „1“ je veoma pouzdana. Također, vrijedi li $y \ll 0$, odluka o klasifikaciji u grupu „-1“ je veoma pouzdana.

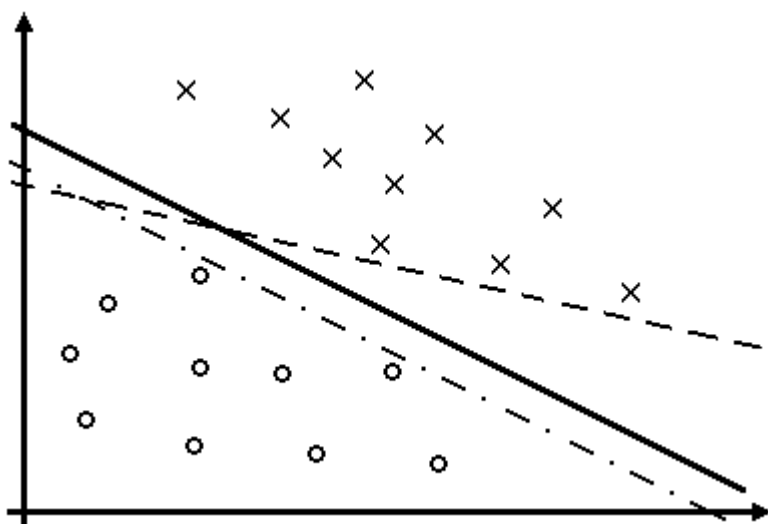
Intuitivno je jasno da je poželjno imati takvu funkciju odluke o klasifikaciji $f(x)$ da za cijeli dani skup za učenje vrijedi:

$$y^{(i)} = 1 \Rightarrow f(x^{(i)}) \gg 0 \wedge y^{(i)} = -1 \Rightarrow f(x^{(i)}) \ll 0,$$

što znači da su sve odluke o klasifikaciji dane funkcije pouzdane.

1.1.2. Intuicija o najboljoj granici odluke

Promatra se binarna klasifikacija s grupama „1“ i „-1“. Svaki objekt se za potrebe klasifikacije predstavlja kao vektor značajki (eng. *Feature vector*). Vektor značajki $x \in \mathbf{R}^n$ sadrži n realnih brojeva koji predstavljaju pojedine osobine objekta (npr. cijena objekta, veličina, površina koju zauzima, težina i slično). Budući da je $x \in \mathbf{R}^n$, vektor značajki se može promatrati kao točka u n -dimenzionalnom prostoru. Pretpostavlja se da su grupe „1“ i „-1“ linearno odvojive u n -dimenzionalnom prostoru. To znači da postoji hiperravnina koja dijeli n -dimenzionalni prostor na dva djela tako da se svi elementi (točke) grupe „1“ nalaze s jedne strane hiperravnine, a svi elementi (točke) grupe „-1“ nalaze s druge strane hiperravnine. U slučaju $n = 2$ dobivamo 2D prostor, a hiperravnina postaje pravac.



Slika 1.1. Linearno odvojive grupe u 2D prostoru

Na slici 1.1 su prikazane dvije linearno odvojive grupe (grupa „1“ je označena kružićem, a grupa „-1“ iksićem). Hiperravnina koja odvaja grupe je pravac. Na slici 1.1 su prikazana tri pravca koja uspješno odvajaju grupe. Od tri prikazana odvajajuća pravca najbolji je pravac prikazan punom linijom. Taj je pravac najudaljeniji od obje grupe, te time smanjuje mogućnost pogrešne klasifikacije u rubnom području grupa, uz sam pravac. To je druga intuicija: najbolji odvajajući pravac dvije linearno odvojive grupe je onaj koji je najudaljeniji od obje grupe.

1.2. Klasifikator optimalne granice

Stroj s potpornim vektorima (eng. *Support Vector Machine*, SVM) se temelji na starijem Klasifikatoru optimalne granice (eng. *Optimal Margin Classifier*, poznat i pod nazivom *Maximum Margin Classifier*), te će prije samog SVM-a biti objašnjen ovaj algoritam.

1.2.1. Korištena notacija

Da bi bilo moguće matematički postaviti temelje algoritma potrebno je uvesti određenu notaciju.

Neka je $y \in \{-1, +1\}$ oznaka pojedine grupe i neka je definirana funkcija

$$g(z) = \begin{cases} 1, & \text{ako } z \geq 0 \\ -1, & \text{ako je } z < 0 \end{cases}$$

Nadalje, neka je $h(x)$ hipoteza koja vraća oznaku grupe (+1 ili -1). Hipoteza $h(x)$ je parametrizirana s parametrima w i b . Pri tome je $x \in \mathbf{R}^n$ vektor značajki, a $w \in \mathbf{R}^n$ i $b \in \mathbf{R}$ su parametri hipoteze.

Vrijedi: $h_{w,b}(x) = g(w^T x + b)$

Skup za učenje je skup primjera za učenje koji su predstavljeni kao uređeni parovi $(x^{(i)}, y^{(i)})$ gdje je $x^{(i)} \in \mathbf{R}^n$ vektor značajki i-tog objekta, a $y^{(i)}$ je oznaka grupe kojoj pripada i-ti objekt.

1.2.2. Funkcijska i geometrijska margina

Par parametara (w, b) definira klasifikator, tako što definira određenu razdvajajuću hiperravninu koja razdvaja grupe objekata.

Funkcijska margina

Definicija : Funkcijska margina hiperravnine (w, b) s obzirom na neki određeni primjer za učenje $(x^{(i)}, y^{(i)})$ je $\hat{\gamma}^{(i)} = y^{(i)}(w^T x^{(i)} + b)$.

Ovako definirana funkcijska margina zapravo matematički iznosi intuiciju o pouzdanosti klasifikacije. Naime, ako je $y^{(i)} = 1$, da bi funkcijska margina bila što veća, mora vrijediti

$w^T x^{(i)} = 0$, odnosno ako je $y^{(i)} = -1$, mora vrijediti $w^T x^{(i)} = 0$. Što je veća funkcijska margina, to je veća i pouzdanost klasifikacije. Vrijedi da je i -ti primjer za učenje $(x^{(i)}, y^{(i)})$ ispravno klasificiran vrijedi li $\hat{\gamma}^{(i)} > 0$, što slijedi direktno iz same definicije.

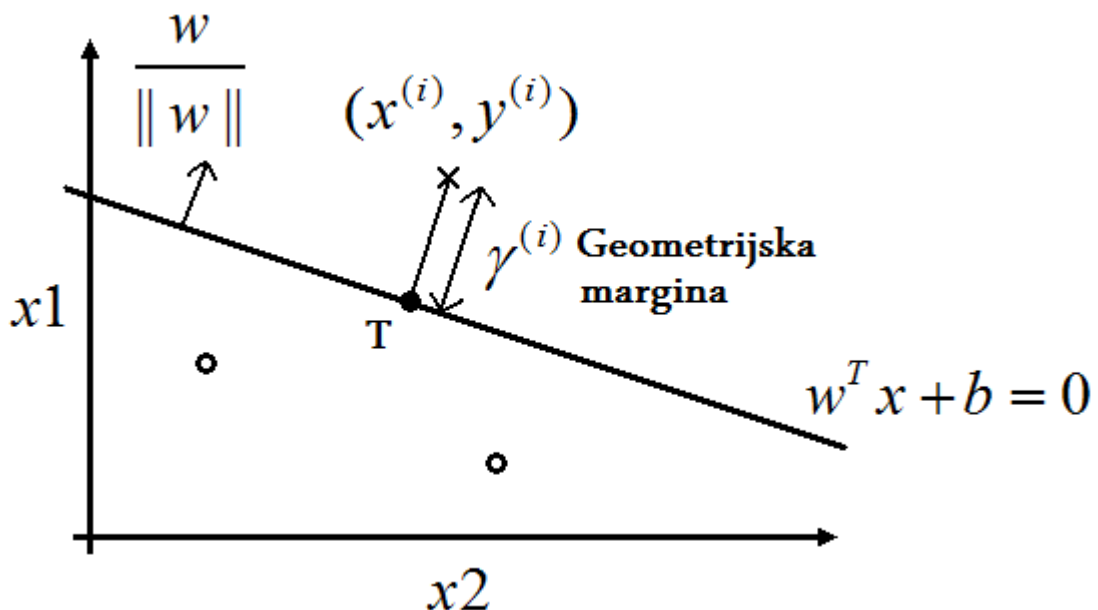
Definicija: Funkcijska margina hiperravnine (w, b) s obzirom na cijeli skup za učenje je $\hat{\gamma} = \min_i \hat{\gamma}^{(i)}$.

Odnosno, funkcijska margina s obzirom na cijeli skup za učenje je najlošiji slučaj funkcijske margine s obzirom na neki određeni primjer za učenje iz danog skupa za učenje.

Rečeno je kako je poželjno imati što veću funkcijsku marginu. Problem s tim zahtjevom je taj da je funkcijsku marginu moguće povećati za željeni faktor tako da se za taj faktor skaliraju parametri w i b (npr. $(w \rightarrow 2w \wedge b \rightarrow 2b) \Rightarrow \hat{\gamma} \rightarrow 2\hat{\gamma}$). Ovaj problem je moguće riješiti dodavanjem normalizacijskog uvjeta, npr. ako se zahtjeva da vrijedi $\|w\| = 1$.

Geometrijska margina

Definicija: Geometrijska margina $\gamma^{(i)}$ hiperravnine (w, b) s obzirom na neki određeni primjer za učenje $(x^{(i)}, y^{(i)})$ je geometrijska udaljenost točke koja predstavlja taj primjer za učenje od hiperravnine (w, b) .



Slika 1.2. Geometrijska margina u 2D

Na slici 1.2 je prikazana geometrijska margina $\gamma^{(i)}$ u 2D prostoru određenom s x_1 i x_2 osima. Hiperravnina je zadana s $w^T x + b = 0$, pa je normala na tu hiperravninu dana s $\vec{n} = \frac{w}{\|w\|}$. Točka T je dana s $x^{(i)} - \gamma^{(i)} \frac{w}{\|w\|}$. Kako točka T leži na hiperravnini, mora zadovoljavati jednadžbu hiperravnine, pa vrijedi:

$$w^T \left(x^{(i)} - \gamma^{(i)} \frac{w}{\|w\|} \right) + b = 0$$

$$w^T x^{(i)} + b = \gamma^{(i)} \frac{w^T w}{\|w\|} = \gamma^{(i)} \|w\|$$

$$\gamma^{(i)} = \left(\frac{w}{\|w\|} \right)^T x^{(i)} + \frac{b}{\|w\|}$$

čime je izražena udaljenost primjera za učenje $(x^{(i)}, y^{(i)})$ od hiperravnine zadane s (w, b) .

Uzme li se u obzir mogućnost da primjer za učenje nije ispravno klasificiran, geometrijska je margina dana s: $\gamma^{(i)} = y^{(i)} \left[\left(\frac{w}{\|w\|} \right)^T x^{(i)} + \frac{b}{\|w\|} \right]$.

Dobiveni izraz za geometrijsku marginu identičan je izrazu za funkcijsku marginu, samo je još i normaliziran s $\|w\|$ (tj. $\gamma^{(i)} = \frac{\hat{\gamma}^{(i)}}{\|w\|}$). Cilj je imati što veću geometrijsku marginu.

Odnosno, ako je objekt ispravno klasificiran, bolje je da je što dalje od odvajajuće hiperravnine.

Definicija: Geometrijska margina hiperravnine (w, b) s obzirom na cijeli skup za učenje je $\gamma = \min_i \gamma^{(i)}$.

1.2.3. Formulacija optimizacijskog problema

Klasifikator optimalne granice je algoritam koji bira parametre w i b tako da maksimizira geometrijsku marginu.

Algoritam postavlja sljedeći optimizacijski problem:

$$\max_{\gamma, w, b} \gamma, \text{ pod uvjetom da vrijedi: } y^{(i)} (w^T x^{(i)} + b) \geq \gamma \wedge \|w\| = 1$$

Budući da se geometrijska margina ne mijenja u ovisnosti o $\|w\|$, moguće je postaviti $\|w\|$ na proizvoljnu vrijednost. U ovoj formulaciji optimizacijskog problema zahtjeva se $\|w\|=1$ što izjednačava geometrijsku marginu s funkcijskom marginom.

Problem trenutne formulacije optimizacijskog problema je što $\|w\|=1$ nije konveksno ograničenje (zahtjeva da w leži na jediničnoj hiperkugli).

Poželjno je naći formulaciju optimizacijskog problema koja ima samo konveksna ograničenja, jer tada će postojati samo globalni optimum, pa će ispravno raditi bilo koji algoritam lokalne pretrage, poput Gradijentskog spusta.

Optimizacijski problem moguće je reformulirati na sljedeći način:

$$\max_{\hat{\gamma}, w, b} \frac{\hat{\gamma}}{\|w\|}, \text{ pod uvjetom da vrijedi: } y^{(i)}(w^T x^{(i)} + b) \geq \hat{\gamma}$$

I dalje je cilj maksimizirati geometrijsku marginu (jer $\gamma^{(i)} = \frac{\hat{\gamma}^{(i)}}{\|w\|}$). Na ovaj način nema više problematičnog nekonveksnog ograničenja, no zato je sada funkcija cilja postala nekonveksna, te ne postoje garancije potrebne za primjenu algoritama lokalne pretrage.

Budući da je moguće koristiti proizvoljno ograničenje na skaliranje w i b , neka je ograničenje $\hat{\gamma}=1$, odnosno $y^{(i)}(w^T x^{(i)} + b) = 1$. Tada se iz $\max_{\hat{\gamma}, w, b} \frac{\hat{\gamma}}{\|w\|} = \max_{w, b} \frac{1}{\|w\|}$ dobije

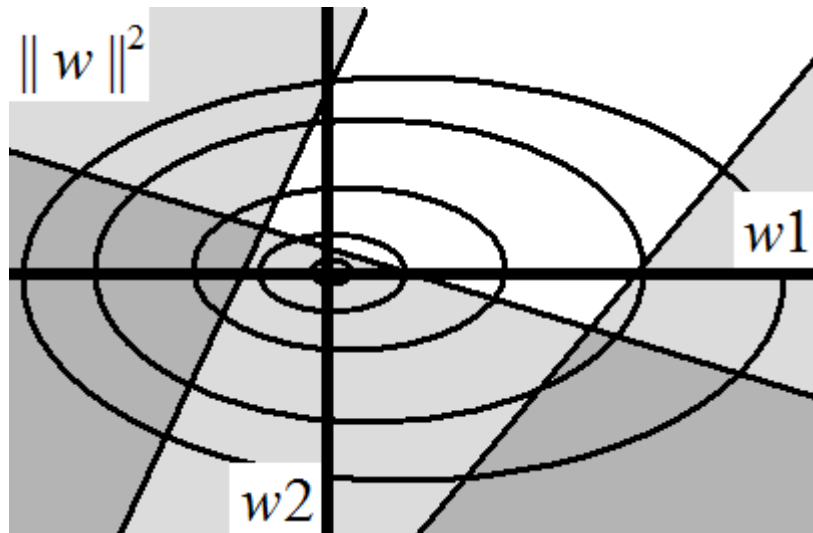
$$\min_{w, b} \|w\|^2.$$

Konačna formulacija optimizacijskog problema:

$$\min_{w, b} \|w\|^2, \text{ pod uvjetom da vrijedi: } y^{(i)}(w^T x^{(i)} + b) \geq 1$$

U konačnoj formulaciji je funkcija cilja kvadratna funkcija (dakle konveksna), a ograničenja su linearna ograničenja na parametre koja izbacuju poluprostore kao nedozvoljene.

Slika 1.3 prikazuje funkciju cilja i ograničenja u konačnoj formulaciji optimizacijskog problema.



Slika 1.3. Funkcija cilja i ograničenja u konačnoj formulaciji optimizacijskog problema

Dobivenu formulaciju problema moguće je riješiti modifikacijom Gradijentskog spusta, no još je bolje ubaciti ovu formulaciju u neki *Quadratic programming (QP) Solver* i time je riješen problem Klasifikatora optimalne granice.

1.2.4. Primal i Dual optimizacijski problem

Ovdje je moguće stati, jer je problem zapravo riješen. Ipak, ovaj optimizacijski problem ima određena svojstva zbog kojih je moguće izvesti efikasnije rješenje koje će kasnije biti modificirano tako da nastane Stroj s potpornim vektorima.

Lagrangeovi multiplikatori

Pomoću metode Lagrangeovih multiplikatora moguće je riješiti problem minimizacije $\min_w f(w)$, pod uvjetom da je zadovoljen skup ograničenja: $h_i(w) = 0$, $i = 1, \dots, l$.

Konstruira se Lagrangian: $L(w, \beta) = f(w) + \sum_{i=1}^l \beta_i h_i(w)$, pri čemu su parametri β_i

Lagrangeovi multiplikatori.

Problem se rješava tako da se $\frac{\partial L}{\partial w}$ postavi na 0, i da se $\frac{\partial L}{\partial \beta}$ postavi na 0. Da bi w^* bilo

rješenje, mora vrijediti da: $\exists \beta^*$, takav da $\frac{\partial L(w^*, \beta^*)}{\partial w} = 0 \wedge \frac{\partial L(w^*, \beta^*)}{\partial \beta^*} = 0$

Primal problem

Optimizacijski problem Klasifikatora optimalne granice se može riješiti primjenom generaliziranih Lagrangeovih multiplikatora.

Neka je potrebno riješiti optimizacijski problem zadan s:

$$\min_w f(w), \text{ tako da bude zadovoljeno: } g_i(w) \leq 0, i = 1, \dots, k \text{ i } h_i(w) = 0, i = 1, \dots, l$$

Konstruiraj se Generalizirani Lagrangian: $L(w, \alpha, \beta) = f(w) + \sum_{i=1}^k \alpha_i g_i(w) + \sum_{i=1}^l \beta_i h_i(w)$

Neka je $\theta_p(w) = \max_{\substack{\alpha, \beta \\ \alpha_i \geq 0}} L(w, \alpha, \beta)$ i neka je $p^* = \min_w \theta_p(w) = \min_w \max_{\substack{\alpha, \beta \\ \alpha_i \geq 0}} L(w, \alpha, \beta)$,

Oznaka „p“ stoji za „primal problem“. Vrijedi: $\theta_p(w) = \begin{cases} +\infty, & \text{ako } g_i(w) > 0 \\ +\infty, & \text{ako } h_i(w) \neq 0 \\ f(w), & \text{inače} \end{cases}$

Vrijedi li $g_i(w) > 0$, narušeno je ograničenje optimizacije. Postavljanjem $\alpha_i = +\infty$ dobije se $\theta_p(w) = +\infty$. Isto vrijedi i za $h_i(w) \neq 0$, jer je opet narušeno ograničenje optimizacije.

Ako nisu narušena ograničenja optimizacije, sumacije u $L(w, \alpha, \beta)$ su jednake 0, jer se maksimum postiže kad su svi $\alpha_i = 0$ i $\beta_i = 0$, stoga je $\theta_p(w) = f(w)$.

Zato je: $\theta_p(w) = \begin{cases} +\infty, & \text{ako su ograničenja narušena} \\ f(w), & \text{inače} \end{cases}$

To znači da je $p^* = \min_w \theta_p(w)$ zapravo originalni problem.

Dual problem

Neka je $\theta_d(\alpha, \beta) = \min_w L(w, \alpha, \beta)$, neka je $d^* = \max_{\substack{\alpha, \beta \\ \alpha_i \geq 0}} \theta_d(\alpha, \beta) = \max_{\substack{\alpha, \beta \\ \alpha_i \geq 0}} \min_w L(w, \alpha, \beta)$,

gdje oznaka „d“ stoji za „dual problem“. Razlika između primal i dual problema, između p^* i d^* , je jedino u poretku maksimizacije i minimizacije.

Uvijek vrijedi $d^* \leq p^*$ (tj. $\max \min(\dots) \leq \min \max(\dots)$).

Pod određenim uvjetima vrijedi $d^* = p^*$. Tada je moguće riješiti dual problem umjesto primal problema.

Karush–Kuhn–Tucker uvjeti

Neka su f i g_i konveksne funkcije ($Hessian \geq 0$), neka su h_i affine funkcije ($h_i(w) = a_i^T w + b_i$), neka su g_i takvi da $\exists w$ za koji vrijedi $g_i(w) < 0$ (stroga nejednakost).

Tada $\exists w^*, \alpha^*, \beta^*$ takvi da: w^* rješava primal problem (p^*), i α^*, β^* rješavaju dual problem (d^*) i da vrijedi $p^* = d^* = L(w^*, \alpha^*, \beta^*)$. Nadalje w^*, α^*, β^* zadovoljavaju Karush–Kuhn–Tucker (KKT) uvjete:

$$\frac{\partial}{\partial w_i} L(w^*, \alpha^*, \beta^*) = 0, \quad i = 1, \dots, n \quad (\text{KKT 1})$$

$$\frac{\partial}{\partial \beta_i} L(w^*, \alpha^*, \beta^*) = 0, \quad i = 1, \dots, l \quad (\text{KKT 2})$$

$$\alpha_i^* g_i(w^*) = 0, \quad i = 1, \dots, k \quad (\text{KKT 3 - KKT complementarity condition})$$

$$g_i(w^*) \leq 0, \quad i = 1, \dots, k \quad (\text{KKT 4})$$

$$\alpha_i^* \geq 0, \quad i = 1, \dots, k \quad (\text{KKT 5})$$

Iz KKT 3 slijedi da barem jedan od $\alpha_i^*, g_i(w^*)$ mora biti jednak nuli.

Zato $\alpha_i^* > 0 \Rightarrow g_i(w^*) = 0$. Obično vrijedi $\alpha_i^* \neq 0 \Leftrightarrow g_i(w^*) = 0$.

Primjena na Klasifikator optimalne granice

Promjena u notaciji za primjenu na Klasifikator optimalne granice:

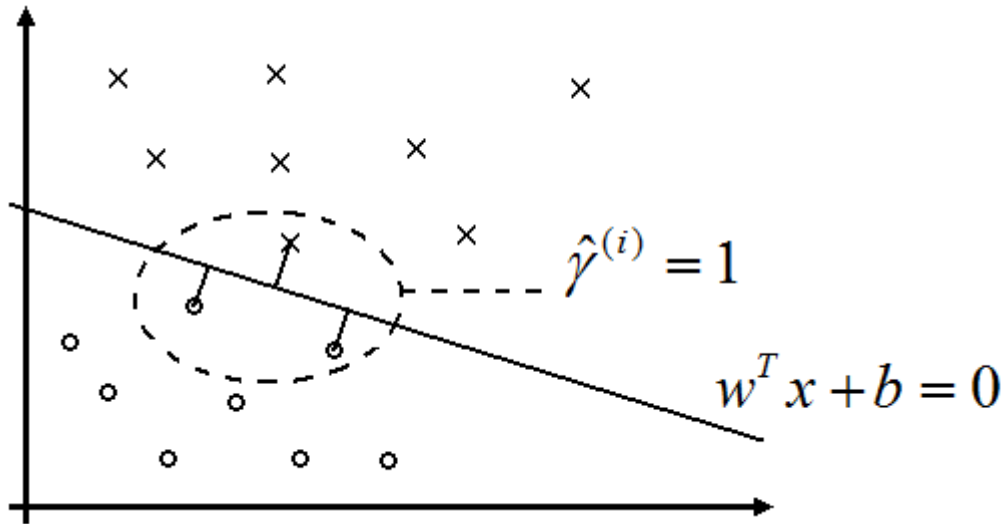
generalizirani Lagrangeovi multiplikatori $\alpha_i, \beta_i \rightarrow \alpha_i$ (postoje samo g_i nejednakosti u optimizacijskom problemu, nema h_i jednakosti), parametri $w \rightarrow w, b$.

Optimizacijski problem:

$$\min_{w, b} \frac{1}{2} \|w\|^2, \quad \frac{1}{2} \text{ dodana radi ljepšeg računa, pod uvjetom da vrijedi } y^{(i)}(w^T x^{(i)} + b) \geq 1$$

$$\text{Uvjet se može napisati kao: } g_i(w, b) = -y^{(i)}(w^T x^{(i)} + b) + 1 \leq 0.$$

Iz KKT 3 vrijedi: $\alpha_i^* > 0 \Rightarrow g_i(w^*) = 0$. Zato $\alpha_i^* > 0 \Rightarrow \hat{y}^{(i)} = 1$, tj. $(x^{(i)}, y^{(i)})$ ima funkcijsku marginu jednakju jedan, što je prikazano na slici 1.4.



Slika 1.4. Potporni vektori

Obično vrijedi $\hat{\gamma}^{(i)} = 1 \Leftrightarrow \alpha_i \neq 0$.

Primjeri za učenje koji su dalje od odvajajuće hiperravnine imaju $\hat{\gamma}^{(i)} > 1$.

Kao što slika 1.4 sugerira, obično primjera za učenje s $\hat{\gamma}^{(i)} = 1$ ima relativno malo (s obzirom na cijeli skup za učenje). Primjeri za učenje s $\hat{\gamma}^{(i)} = 1$ nazivaju se potpornim vektorima (eng. *Support Vectors*). Odavde potječe ime Stroj s potpornim vektorima (eng. *Support Vector Machine*, SVM). Kako je relativno malo potpornih vektora, većinom vrijedi $\alpha_i = 0$, jer je $\alpha_i = 0$ za primjere za učenje koji nisu potporni vektor.

Generalizirani Lagrangian je: $L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^m \alpha_i (y^{(i)} (w^T x^{(i)} + b) - 1)$.

Za dual problem vrijedi: $\theta_d(\alpha) = \min_{w, b} L(w, b, \alpha)$. Potrebno je minimizirati Generalizirani

Lagrangian, odnosno znači izjednačiti parcijalne derivacije s nulom. Dobije se:

$\nabla_w L(w, b, \alpha) = w - \sum_{i=1}^m \alpha_i y^{(i)} x^{(i)} = 0$, zato vrijedi: $w = \sum_{i=1}^m \alpha_i y^{(i)} x^{(i)}$ (tj. w je linearna

kombinacija vektora značajki). Nadalje, $\frac{\partial}{\partial b} L(w, b, \alpha) = -\sum_{i=1}^m \alpha_i y^{(i)} = 0$

Rješenje za w se ubaci u formulu za $L(w, b, \alpha)$ i dobije se:

$L(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^m \alpha_i (y^{(i)} (w^T x^{(i)} + b) - 1)$, gdje je: $w^T w = (\sum_{i=1}^m \alpha_i y^{(i)} x^{(i)})^T (\sum_{i=1}^m \alpha_i y^{(i)} x^{(i)})$

$$\text{Zato: } L(w, b, \alpha) = \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m y^{(i)} y^{(j)} \alpha_i \alpha_j (x^{(i)})^T x^{(j)} - \sum_{i=1}^m \sum_{j=1}^m y^{(i)} y^{(j)} \alpha_i \alpha_j (x^{(i)})^T x^{(j)} + \sum_{i=1}^m \alpha_i$$

$$L(w, b, \alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m y^{(i)} y^{(j)} \alpha_i \alpha_j (x^{(i)})^T x^{(j)}$$

$$\text{Neka je } W(\alpha) = L(w, b, \alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m y^{(i)} y^{(j)} \alpha_i \alpha_j < x^{(i)}, x^{(j)} >$$

$$\text{Dual problem je: } \max_{\alpha} W(\alpha), \text{ uz ograničenja: } \alpha_i \geq 0 \text{ i } \sum_{i=1}^m \alpha_i y^{(i)} = 0$$

$$\text{Drugo ograničenje dolazi iz } \frac{\partial}{\partial b} L(w, b, \alpha) = 0$$

Da nema tog ograničenja bilo bi $\theta_d(\alpha) = -\infty$, jer je $L(w, b, \alpha)$ linearna funkcija parametra

$$b: L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^m \alpha_i (y^{(i)} (w^T x^{(i)} + b) - 1), \text{ a provodi se minimizacija.}$$

S tim ograničenjem vrijedi: $\theta_d(\alpha) = W(\alpha)$.

Postupak rješavanja problema Klasifikatora optimalne granice

Prvo se riješi dual problem za α ,

$$\text{zatim se odredi } w \text{ iz izraza } w = \sum_{i=1}^m \alpha_i y^{(i)} x^{(i)},$$

$$\text{te je još potrebno odrediti } b \text{ iz } b = - \frac{\max_{i: y^{(i)} = -1} w^T x^{(i)} + \min_{i: y^{(i)} = 1} w^T x^{(i)}}{2}.$$

Interpretacija formule za b : nađe se najgori primjer za učenje iz grupe „1“ i najgori primjer za učenje iz grupe „-1“, te se stavi hiperravnina na pola njihove međusobne udaljenosti.

Opisani postupak ima dobro svojstvo da samo potporni vektori imaju $\alpha_i \neq 0$, zbog čega je potrebno pamtiti samo potporne vektore jednom kada je klasifikator izgrađen.

Radi potpunosti treba spomenuti da su uvjeti potrebni da vrijedi $d^* = p^*$ (i ispravnost KKT uvjeta) ispunjeni, pa je sav provedeni postupak ispravan.

1.3. Jezgreni trik

Klasifikator optimalne granice je prilično dobar linearni klasifikator. Ipak, baš zato što je linearan, ima ograničenu primjenu jer je primjenjiv samo na klasifikaciju linearno odvojivih grupa objekata (u odnosu na izabrani skup značajki). Stroj s potpornim vektorima je modificirani Klasifikator optimalne granice koji nije ograničen samo na linearnu klasifikaciju. Mogućnost primjene na linearno neodvojive grupe objekata postiže se metodom poznatom pod nazivom Jezgreni trik (eng. *Kernel Trick*, [3]).

1.3.1. Motivacija

Postoji cijeli niz problema koje je moguće izraziti (točnije, koji su vezani uz podatke) samo preko unutarnjeg produkta (eng. *Inner product*), umnoška dva vektora koji kao rezultat daje realan broj. Klasifikator optimalne granice takav je problem.

Hipoteza je dana s $h_{w,b}(x) = g(w^T x + b)$, gdje je $w = \sum_{i=1}^m \alpha_i y^{(i)} x^{(i)}$.

Vrijedi $w^T x + b = \sum_{i=1}^m \alpha_i y^{(i)} \langle x^{(i)}, x \rangle + b$, pri čemu je $\langle x^{(i)}, x \rangle = (x^{(i)})^T x$ oznaka za unutarnji produkt vektora $x^{(i)}$ i x .

Također vrijedi $W(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle$, gdje je opet $\langle x^{(i)}, x^{(j)} \rangle = (x^{(i)})^T x^{(j)}$ oznaka za unutarnji produkt vektora $x^{(i)}$ i $x^{(j)}$.

Zato je moguće postupak izgradnje klasifikatora (određivanje w i b), kao i postupak klasifikacije novog objekta povezati s podacima koristeći samo unutarnji produkt. Slijedi da nikada nije potrebno vektore značajki $x^{(i)}$ odnosno x izraziti direktno.

1.3.2. Ideja

Neka je dan vektor značajki $x \in \mathbf{R}^n$, i neka je dana funkcija $\phi: \mathbf{R}^n \mapsto \mathbf{R}^m$, gdje je $m > n$.

Funkcija $\phi(x)$ preslikava točku (vektor značajki) x iz n -dimenzionalnog vektorskog prostora u m -dimenzionalni vektorski prostor, gdje su n - i m -dimenzionalni prostori tzv. Hilbertovi prostori (eng. *Hilbert space*). Pri tome m -dimenzionalni prostor obično ima

puno više dimenzija, a čak je dozvoljeno da je $\phi(x) \in \mathbf{R}^\infty$, tj. da se preslikava u beskonačno-dimenzionalan prostor.

Budući da se cijeli algoritam može izraziti pomoću unutarnjeg produkta, zamjenom $\langle x^{(i)}, x^{(j)} \rangle$ s $\langle \phi(x^{(i)}), \phi(x^{(j)}) \rangle$ cijeli algoritam sada radi s m-dimenzionalnim vektorima značajki $\phi(x)$.

Javlja se problem efikasnog računanja unutarnjeg produkta vektora koji imaju veoma velik (ili čak beskonačan) broj značajki.

Srećom, u mnogim bitnim specijalnim slučajevima moguće je napisati tzv. Jezgenu funkciju (eng. *Kernel Function*) $K(x, z)$, tako da vrijedi $K(x, z) = \langle \phi(x), \phi(z) \rangle$ i da je pri tome moguće izračunati $K(x, z)$ puno efikasnije nego da se eksplicitno računa $\langle \phi(x), \phi(z) \rangle$.

Primjer

Neka je $x, z \in \mathbf{R}^n$ i neka je $K(x, z) = (x^T z)^2$, tada za $K(x, z)$ vrijedi:

$$K(x, z) = (x^T z)^2 = \left(\sum_{i=1}^n x_i z_i \right) \left(\sum_{j=1}^n x_j z_j \right) = \sum_{i=1}^n \sum_{j=1}^n (x_i x_j) (z_i z_j)$$

Nadalje, neka je $\phi(x) = \begin{bmatrix} x_1 x_1 \\ x_1 x_2 \\ \vdots \\ x_n x_{n-1} \\ x_n x_n \end{bmatrix}$. Vrijedi: $K(x, z) = (\phi(x))^T \phi(z)$

Računa li se $K(x, z)$ kao $K(x, z) = (\phi(x))^T \phi(z)$, složenost je $O(n^2)$, a ako se $K(x, z)$ računa kao $K(x, z) = (x^T z)^2$, tada je složenost $O(n)$.

1.3.3. Izbor jezgre

Da bi se izgradio kvalitetan klasifikator, izbor jezgre je prilično bitan.

Neka su x i z vektori značajki, te $x \mapsto \phi(x)$ i $z \mapsto \phi(z)$ slike dobivene jezgrom.

Ako su x i z jako slični, tada su vektori $\phi(x)$ i $\phi(z)$ usmjereni u približno istom smjeru, pa je njihov unutarnji produkt velik, a ako su x i z veoma različiti, tada je njihov unutarnji produkt malen. Ova intuicija je korisna, a nije stroga.

Pri susretu s novim klasifikacijskim problemom jedan od načina kako početi je pokušati naći funkciju $K(x, z)$ koja za objekte koji se promatraju kao slični vraća veliku vrijednost, a za objekte koji se promatraju kao različiti vraća malu vrijednost.

Ispravnost jezgre

Pitanje ispravnosti jezgre je ekvivalentno pitanju postoji li funkcija $\phi(x)$ koja preslikava vektor iz jednog prostora u drugi.

Odnosno, $K(x, z)$ je ispravna jezgra akko $\exists \phi$ takav da vrijedi: $K(x, z) = \langle \phi(x), \phi(z) \rangle$

Mercerov teorem (nužan i dovoljan uvjet da bi $K(x, z)$ bila ispravna jezgra):

Neka je dan proizvoljan skup od $m < \infty$ točaka $\{x_1, \dots, x_m\}$. Jezgra $K(x, z)$ je ispravna (Mercerova) jezgra ako i samo ako za pripadajuću matricu $K \in \mathbf{R}^{n \times n}$, definiranu kao $K_{ij} = K(x^{(i)}, x^{(j)})$ vrijedi da je simetrična i da je $K \geq 0$.

1.3.4. Primjeri jezgara

Kako je izbor jezgre veoma bitan element izgradnje Stroja s potpornim vektorima, slijede neke često korištene jezgre ([4]).

Polinomna jezgra

Generalizacijom jezgre iz prethodno navedenog primjera dobije se tzv. Polinomna jezgra (eng. *Polynomial kernel*): $K(x, z) = (x^T z + c)^d$, gdje je $c \in \mathbf{R}$ i $d \in \mathbf{N}$.

Polinomna jezgra implicitno koristi $\binom{n+d}{d}$ monoma do stupnja d , čime koristi izrazito velik broj značajki, a i dalje se može izračunati u složenosti $O(n)$.

Bitno svojstvo ove jezgre je da čuva i originalne značajke koje su doduše pomnožene konstantom, pa time nužno može linearno razdvojiti u višem prostoru sve što je originalno linearno razdvojivo u prostoru početnih značajki.

Loše svojstvo Polinomne jezgre je to što može vratiti izrazito veliku vrijednost za unutarnji produkt, pa je potrebno provesti normalizaciju početnih značajki prije računalne upotrebe.

Gaussova jezgra

Gaussova jezgra (eng. *Gaussian kernel*) je vjerojatno jezgra s najraširenijom primjenom. Definirana je formulom $K(x, z) = \exp(-\tau \|x - z\|^2)$, gdje je $\tau \in \mathbf{R}$.

Naziva se Gaussovom jezgrom zbog sličnosti formule jezgre s formulom Gaussove krivulje.

Ova jezgra odgovara preslikavanju u $\mathbf{R}^\infty$ prostor, tj. dobije se beskonačno-dimenzionalan prostor.

Sve što je originalno linearno razdvojivo u prostoru početnih značajki, razdvojivo je i u višem prostoru određenom ovom jezgrom.

Za velik broj problema je Gaussova jezgra dobar izbor početne jezgre.

Racionalna kvadratna jezgra

Racionalna kvadratna jezgra (eng. *Rational quadratic kernel*) je definirana formulom:

$$K(x, z) = 1 - \frac{\|x - z\|^2}{\|x - z\|^2 + \tau}, \text{ gdje je } \tau \in \mathbf{R}.$$

Ova jezgra je relativno dobra zamjena za Gaussovu jezgru ako je potrebno izbjeći potenciranje ($\exp(-\tau \|x - z\|^2)$) prisutno u Gaussovoj jezgri.

Primjer nestandardne jezgre – jezgra za klasifikaciju proteina

Neka je potrebno klasificirati proteine u grupe. Proteini su izgrađeni od nizova aminokiselina. Kako aminokiseline određuju svojstva proteina, vektor značajki pojedinog proteina može biti niz aminokiselina. Problem s predstavljanjem konkretnog proteina preko nizova aminokiselina je u tome što su proteini koji spadaju u istu grupu (prema odabranom kriteriju) često višestruko različite duljine. Potrebno je dakle kreirati funkciju $\phi(x)$ koja će dati vektor značajki koji će biti iste duljine za sve proteine, a da se pri tome taj vektor značajki temelji na pripadnom nizu aminokiselina.

Moguće je konstruirati vektor $\phi(x)$ na sljedeći način:

Neka je svaka aminokiselina predstavljena svojim slovom, što je moguće, jer ima 20 aminokiselina i 26 slova (engleske abecede). Neka su slova A-T iskorištena za pripadne aminokiseline.

Neka vektor $\phi(x)$ broji koliko se puta pojavila određena kombinacija od po četiri slova. Time se svaki protein predstavlja vektorom $\phi(x)$ jednake duljine ($20^4 = 160000$).

Ovaj zapis vektora značajki je visoko-dimenzionalan (160000 dimenzija), pa je poželjno naći jezgru koja preslikava izvorni niz aminokiselina u ovakav vektor značajki.

Upotrebom Dinamičkog programiranja može se doći do efikasnog rješenja za izgradnju takve jezgre.

Tako se dobije usko specijalizirana jezgra koja se može relativno efikasno izračunati.

1.4. Stroj s potpornim vektorima L1 meke granice

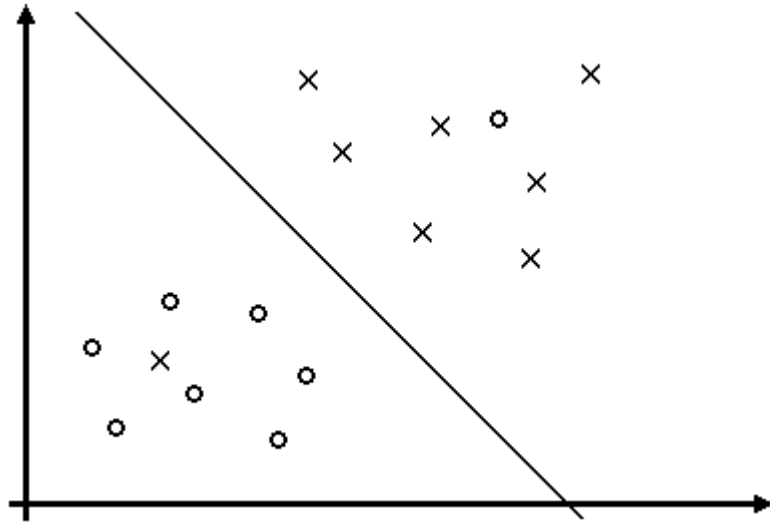
U nastavku je opisana modifikacija Stroja s potpornim vektorima kojom se postižu bolje performanse i veća upravljivost izgradnjom klasifikatora.

1.4.1. Motivacija

Stroj s potpornim vektorima je Klasifikator optimalne granice na koji je primijenjen Jezgreni trik. Zato je u Klasifikatoru optimalne granice svaka pojava unutarnjeg produkta zamijenjena s jezgrenom funkcijom ($\langle x, z \rangle \rightarrow K(x, z)$). Upotrebom jezgrenog trika, Stroj s potpornim vektorima može dati nelinearnu granicu između dvije grupe, pa je primjenjiv i na linearno neodvojive grupe objekata (u odnosu na izabrani skup značajki). Zapravo, Stroj s potpornim vektorima i dalje rezultira linearnom granicom, no ta je granica linearna u prostoru u koji su preslikani vektori značajki funkcijom $\phi(x)$. Funkcija $\phi(x)$ često nije eksplicitno poznata, a ponekad ni sam prostor, u koji se preslikava upotrebom jezgre, nije poznat.

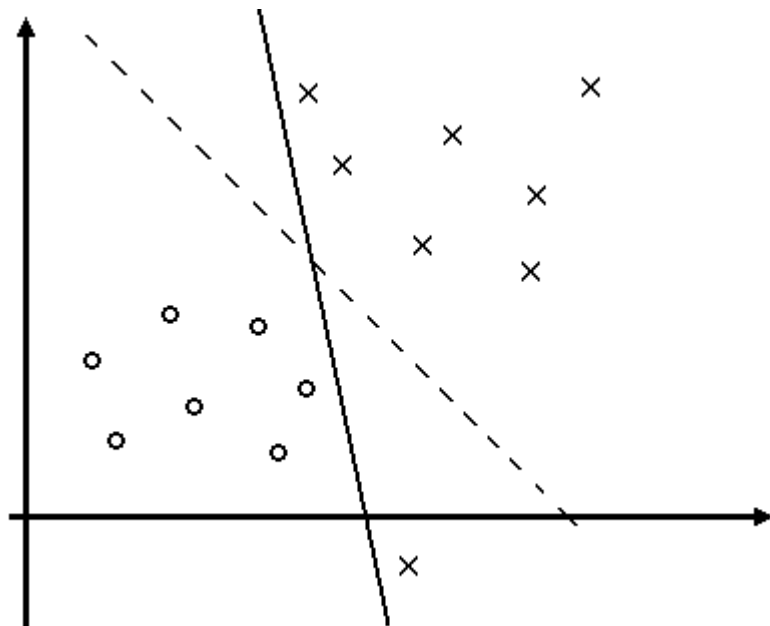
Ovakav Stroj s potpornim vektorima i dalje zahtjeva da su grupe objekata potpuno odvojive (u suprotnom algoritam neće uspješno završiti). Vjerojatnost da su grupe odvojive u nekom prostoru raste s brojem dimenzija. Unatoč tome, upotreba određene jezgre ne garantira da će dani skup za učenje biti (linearno) odvojiv u prostoru određenom s upotrijebljenom jezgrom. Iz tog je razloga korisno modificirati Stroj s potpornim vektorima tako da radi i ako je određeni dio skupa za učenje linearno neodvojiv, čak i u

prostoru određenom upotrebom jezgre. Na slici 1.5 su prikazane dvije linearno neodvojive grupe objekata. Iako grupe nisu odvojive, ipak postoji granica koja odvaja grupe relativno uspješno.



Slika 1.5. Linearno neodvojive grupe objekata

Osim iznesenog problema, javlja se i problem prevelikog utjecaja pojedinog primjera za učenje na resultantnu granicu grupa. Obično je bolje donekle smanjiti utjecaj pojedinog primjera za učenje. Time se izbjegava i slučaj kada je u skupu za učenje neki objekt greškom smješten u krivu grupu. Slika 1.6 ilustrira ovaj problem.



Slika 1.6. Problem prevelikog utjecaja pojedinog primjera za učenje

Jedna modifikacija Stroja s potpornim vektorima koja rješava prethodno izložene probleme naziva se Stroj s potpornim vektorima L1 meke granice (eng. *L1 norm softmargin Support Vector Machine*, [5]).

1.4.2. Modifikacija

Modificira se formulacija problema Stroja s potpornim vektorima:

$$\min_{w,b,\xi_i} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i \text{ pod uvjetom da vrijedi } y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i \text{ i } \xi_i \geq 0, i=1, \dots, m,$$

gdje su ξ_i „kaznene“ varijable.

Ako je $y^{(i)}(w^T x^{(i)} + b) > 0$, tada je objekt ispravno klasificiran, u suprotnom je pogrešno klasificiran. Postavi li se neki $\xi_i > 1$, tada je $\hat{y} < 0$, pa se dozvoljava da primjer iz skupa za učenje bude krivo klasificiran. Ipak, takav izbor se obeshrabruje jer u (minimizacijskom) optimizacijskom cilju stoji: $+C \sum_{i=1}^m \xi_i$, pri čemu zadani parametar C određuje koliko se lošim smatra izbor $\xi_i > 1$.

Izvođenjem dual problema se u konačnici dobije:

$$\max_{\alpha} W(\alpha) = \max_{\alpha} \left[\sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle \right],$$

$$\text{uz ograničenja: } \sum_{i=1}^m \alpha_i y^{(i)} = 0 \text{ i } 0 \leq \alpha_i \leq C, i=1, \dots, m.$$

Odnosno, jedina modifikacija je $0 \leq \alpha_i \leq C$ umjesto $\alpha_i \geq 0$.

Iz KKT uvjeta se dobije kriterij konvergencije, kada su α_i konvergirali k globalnom optimumu. Ovaj se kriterij može koristiti u algoritmu koji rješava optimizacijski problem po α_i :

$$\alpha_i = 0 \Rightarrow y^{(i)}(w^T x^{(i)} + b) \geq 1$$

$$\alpha_i = C \Rightarrow y^{(i)}(w^T x^{(i)} + b) \leq 1$$

$$0 \leq \alpha_i \leq C \Rightarrow y^{(i)}(w^T x^{(i)} + b) = 1$$

2. Slijedna minimalna optimizacija

Slijedna minimalna optimizacija (eng. *Sequential Minimal Optimization*, SMO) je jedan od algoritama ([6]) koji rješava optimizacijski problem Stroja s potpunim vektorima.

2.1. Koordinatni uspon

Slijedna minimalna optimizacija je algoritam inspiriran jednostavnijim algoritmom poznatim pod nazivom Koordinatni uspon (eng. *Coordinate Ascent*).

Potrebno je riješiti sljedeći optimizacijski problem: $\max_{\alpha} W(\alpha_1, \alpha_2, \dots, \alpha_m)$ pri čemu nisu zadana nikakva ograničenja.

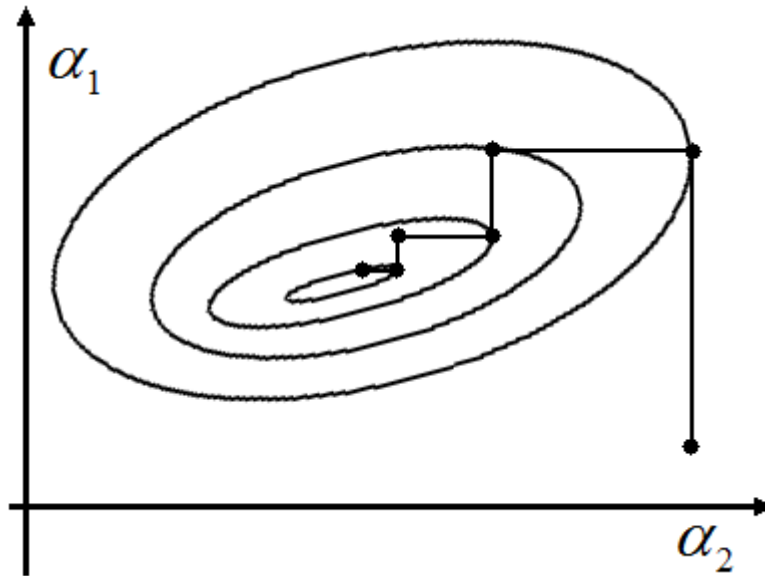
Algoritam koji rješava ovaj problem:

```
Ponavljaj do konvergencije {  
  Za i=1 do m {  
     $\alpha_i = \arg \max_{\hat{\alpha}_i} [W(\alpha_1, \dots, \alpha_{i-1}, \hat{\alpha}_i, \alpha_{i+1}, \dots, \alpha_m)]$   
  }  
}
```

Gdje $\alpha_i = \arg \max_{\hat{\alpha}_i} [W(\alpha_1, \dots, \alpha_{i-1}, \hat{\alpha}_i, \alpha_{i+1}, \dots, \alpha_m)]$ zapravo znači da se maksimizira W , pri čemu se svi α_j , osim upravo promatranog α_i , drže konstantnim tj. maksimizira se W po samo jednom argumentu (α_i).

Slika 2.1 prikazuje kako funkcionira Koordinatni uspon.

Algoritam se postepeno približava rješenju, a svaki se korak optimizacije izvodi duž jedne od koordinatnih osi.



Slika 2.1. Princip rada Koordinatnog uspona

Performanse ovog algoritma se mogu značajno popraviti upotrebom heuristika za izbor sljedećeg parametra po kojem se optimizacija provodi, umjesto da se ciklički redom ide po svim argumentima. Naravno, ovo ima smisla samo u više od dvije dimenzije.

Koordinatni uspon zahtjeva puno više koraka, nego neki drugi algoritmi, da riješi dani optimizacijski problem. Dobra osobina koordinatnog uspona je da je često svaki pojedini korak algoritma izrazito male složenosti, pa u konačnici algoritam ima relativno dobre performanse.

2.2. Primjena na Stroj s potpunim vektorima

Koordinatnim usponom nije moguće riješiti optimizacijski problem Stroja s potpunim vektorima, jer postoje ograničenja na α_i . Nije moguće sve α_i , osim jednog, držati konstantnim u jednoj iteraciji algoritma, zbog ograničenja $\sum_{i=1}^m \alpha_i y^{(i)} = 0$. Ovo ograničenje zahtjeva da je bilo koji α_i u potpunosti određen s preostalim α_j (npr. vrijedi

$\alpha_i = -\frac{\sum_{i \neq 2}^m \alpha_i y^{(i)}}{y^{(1)}}$). Iako nije moguće mijenjati samo jedan α_i , moguće je mijenjati bilo koja

dva α_i , a da se pri tome ne naruše ograničenja na α_i . Kao posljedica ove ideje nastao je algoritam Slijedne minimalne optimizacije (eng. *Sequential Minimal Optimization*, [7]).

U nazivu algoritma je minimalna optimizacija jer se mijenja najmanji mogući broj α_i , a da su pri tome ograničenja zadovoljena.

Nacrt algoritma:

```
Ponavljaj do konvergencije {
    Selektiraj  $\alpha_i, \alpha_j$  pomoću heuristike
    Drži sve  $\alpha_k, k \neq i, j$  konstantnim
    Optimiziraj  $W(\alpha)$  u odnosu na  $\alpha_i, \alpha_j$ 
    (uvažavajući pri tome sva ograničenja)
}
```

Optimizacijski korak je moguće izvesti izrazito brzo, pa je ovaj algoritam veoma efikasan.

Da bi notacija bila što jednostavnija, neka je potrebno optimizirati $W(\alpha)$ u odnosu na α_1, α_2 (provedeno razmatranje je identično je za bilo koji par α_i, α_j).

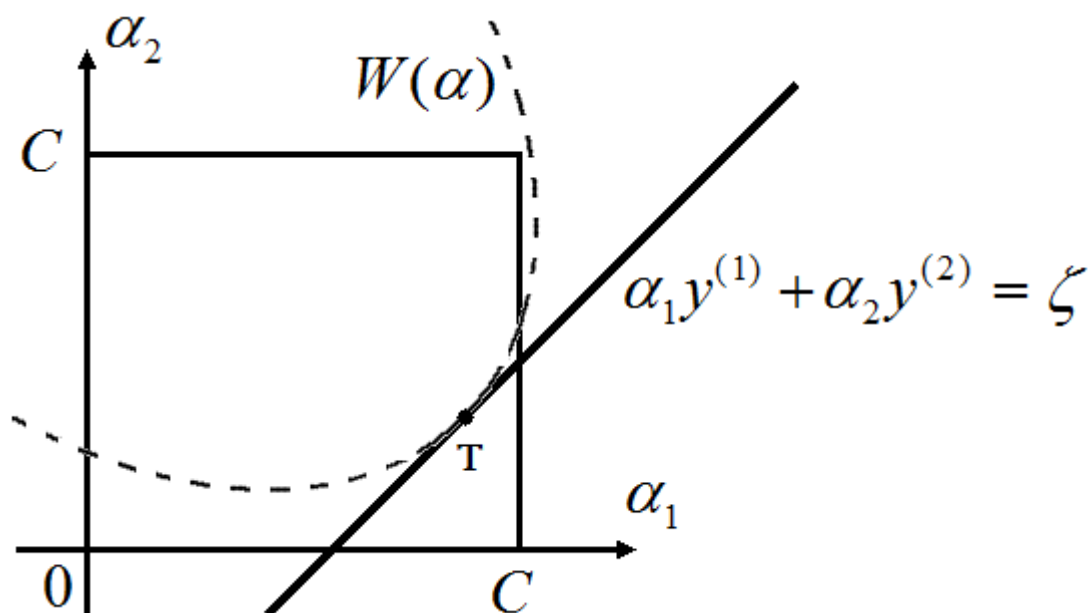
Iz ograničenja $\sum_{i=1}^m \alpha_i y^{(i)} = 0$ odnosno $\alpha_1 y^{(1)} + \alpha_2 y^{(2)} = -\sum_{i=3}^m \alpha_i y^{(i)} = \zeta$ slijedi

$\alpha_1 = \frac{\zeta - \alpha_2 y^{(2)}}{y^{(1)}}$. Postoji još i ograničenje $0 \leq \alpha_i \leq C$. Zato je

$$W(\alpha_1, \alpha_2, \alpha_3, \dots) = W\left(\frac{\zeta - \alpha_2 y^{(2)}}{y^{(1)}}, \alpha_2, \alpha_3, \dots\right).$$

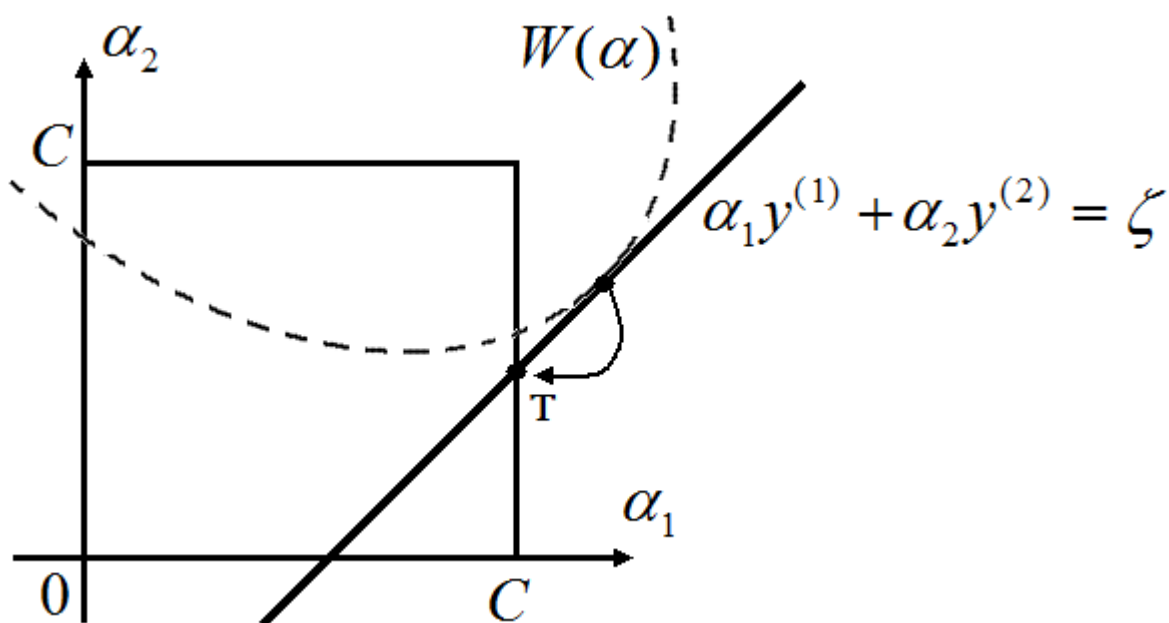
Kako je $W(\alpha)$ kvadratna funkcija, moguće ju je, u kontekstu optimizacije $W(\alpha)$ u odnosu na α_1, α_2 , promatrati kao jednodimenzionalnu kvadratnu funkciju varijable α_2 . To znači da je moguće funkciju $W(\alpha_2)$ zapisati kao $a\alpha_2^2 + b\alpha_2 + c$, što je moguće izrazito efikasno (analitički) optimizirati.

Slika 2.2 prikazuje optimizaciju s obzirom na α_1 i α_2 . Izračunati maksimum optimizacije se postiže u točki T. Kako za točku T vrijedi $0 \leq \alpha_1, \alpha_2 \leq C$, izračunato rješenje optimizacije zadovoljava sva ograničenja.



Slika 2.2. Optimizacija s obzirom na α_1 i α_2 .

Na slici 2.3 prikazana je situacija gdje je ograničenje $0 \leq \alpha_1, \alpha_2 \leq C$ narušeno. Da bi rješenje optimizacije bilo ispravno, potrebno je pomaknuti rješenje od najbliže točke koja zadovoljava sva ograničenja, u ovom slučaju do točke T.



Slika 2.3. Optimizacija s obzirom na α_1 i α_2 uz narušavanje ograničenja

Efikasnost SMO algoritma uvelike ovisi o dobrom odabiru heuristike koja izabire α_i, α_j koji će se sljedeći optimizirati.

Uzastopnom primjenom optimizacijskih koraka algoritam se približava traženom rješenju. Kako se u pojedinom koraku mijenja par α_i, α_j , ponekad se dogodi da se optimizacijom α_i (približavanjem α_i njegovoj konačnoj vrijednosti), α_j udalji od njegovog konačnog rješenja. Unatoč ovom udaljavanju α_j od njegovog rješenja, ukupni učinak svakog pojedinog koraka je pozitivan, pa se algoritam svakim korakom približava traženom rješenju ([8]).

3. Problem raspoznavanja prometnih znakova

Problem raspoznavanja (klasifikacije) prometnih znakova moguće je riješiti Strojem s potpornim vektorima.

3.1. Reprezentacija prometnih znakova

Prometne znakove potrebno je predstaviti na prikladan način da bi bilo moguće provesti postupak klasifikacije. Za potrebe Stroja s potpornim vektorima, potrebno je sve prometne znakove predstaviti jednakim brojem značajki (u originalnom ili jezgrenom prostoru).

3.1.1. Početne fotografije prometnih znakova

Prometni su znakovi početno predstavljeni fotografijama. Fotografije su snimljene tokom vožnje iz vozila koje normalno putuje prometnicama. Zato su fotografije nastale u prilično raznovrsnim uvjetima. Snimane su u različitim vremenskim uvjetima što uzrokuje prilične razlike u osvjetljenju, vidljivosti i slično. Snimane su i u različito doba dana što također uzrokuje znatne razlike u osvjetljenju i kontrastu. Otežavajuća je okolnost i ta što su fotografije snimane iz vozila koje se kreće različitim brzinama. Tako fotografije imaju različit stupanj iskrivljenosti nastale zbog brzine. Toj razlici doprinosi i udaljenost znaka od objektiva, jer su relativna gibanja znakova u slici puno izraženija što je znak bliže objektivu. Kako su znakovi snimani na različitim udaljenostima od objektiva, tako su i različitih relativnih veličina naspram cijele slike.

Iz prethodno navedenog jasno je kako su početno dostupni podatci (fotografije) daleko od savršenih za klasifikaciju, no zato su puno bliži realnoj upotrebi, jer nisu nastali u kontroliranim laboratorijskim uvjetima.

Primjer jedne fotografije s prometnim znakovima je na slici 3.1.



Slika 3.1. Primjer fotografije s prometnim znakovima

3.1.2. Podatci uz fotografije prometnih znakova

Uz originalne fotografije znakova (poput fotografije prikazane na slici 3.1), dostupni su i podaci o položaju znaka u fotografiji. Ti su podaci nastali ručnim označavanjem znakova u fotografijama, a kao rezultat procesa se za svaki znak dobiju koordinate znaka kao i njegova širina i visina. Time je potpuno određen položaj znaka u fotografiji.

Potrebno je napomenuti da su ovi podaci daleko od savršenih za klasifikaciju pomoću Stroja s potpornim vektorima, jer nisu ni nastali s primarnim ciljem da se upotrijebe za ovaj zadatak. Nesavršenost podataka se prvenstveno odražava u raznolikosti u označavanju položaja i veličine prometnog znaka u fotografiji. Varira udio prometnog znaka unutar označenog područja (nekad je označeno izrazito malo područje oko samog znaka, a nekad sam znak čini tek malen dio označenog područja). Problem je i u pouzdanosti označavanja uzrokovanoj ljudskim faktorom. Nekad je znak zahvaćen tek svojim djelom unutar

označenog područja, a nekad čak nije ni prisutan u označenom području, pa je npr. drvo označeno kao da sadrži znak, a ne sadrži ga.

3.1.3. Normalizirane fotografije prometnih znakova

Kako bi se izloženi problemi donekle ublažili i ujednačili, provedena je normalizacija fotografija. Kao rezultat nastale su malene fotografije, dimenzije 24x24 slikovna elementa. Na ovakvim fotografijama je i dalje moguće prepoznati prometni znak, pa zadovoljavaju nužan uvjet potreban za klasifikaciju. Primjeri normaliziranih prometnih znakova dani su na slici 3.2. Primjetna je varijacija u kvaliteti fotografija, kao i razlika u fizičkom ostvarenju prometnih znakova (npr. žuta ili bijela boja). Normalizirane fotografije izrazito loše kvalitete su izuzete iz daljnjeg postupka.



Slika 3.2. Primjeri normaliziranih fotografija prometnih znakova

3.1.4. Izbor vektora značajki

Normaliziranim fotografijama 24x24 slikovna elementa su svi prometni znakovi reprezentirani na isti način. Iz normaliziranih fotografija moguće je odabrati vektore značajki na mnogo različitih načina. Moguće je recimo izgraditi histogram za svaki prometni znak ([9]), pa vrijednosti histograma koristiti kao vektor značajki. Ovaj pristup bi za klasifikaciju prometnih znakova davao prilično loše rezultate, budući da mnogi znakovi imaju identične histograme, poput znaka upozorenja za lijevi zavoj i znaka upozorenja za desni zavoj.

Odabrani vektor značajki sastoji se od redom poredanih vrijednosti slikovnih elemenata u RGB sustavu boja. Točnije, svaki je slikovni element predstavljen u vektoru značajki s tri

broja, po jednim za svaku komponentu boje u RGB sustavu. Tako vektor značajki ima $24 \cdot 24 \cdot 3 = 1728$ elemenata.

Ovakav vektor značajki ima srednje velik broj značajki, dovoljno velik da se može koristiti bez jezgre, a opet, moguće je primijeniti i jezgru pa tako implicitno vršiti klasifikaciju u prostoru s značajno većim brojem dimenzija.

3.2. Programsko ostvarenje Stroja s potpornim vektorima

Stroj s potpornim vektorima je realiziran u programskom jeziku Java. Optimizacijski problem Stroja s potpornim vektorima riješen je algoritmom Slijedne minimalne optimizacije.

3.2.1. Programsko ostvarenje Slijedne minimalne optimizacije

Algoritam Slijedne minimalne optimizacije je programski ostvaren u svom osnovnom obliku, uz relativno malen broj modifikacija.

Upotrijebljena je standardna, preporučena heuristika za odabir para α_i, α_j jer se pokazala dovoljno efikasnom za dani problem. Istraživanja su pokazala ([11]) da postoje i bolje heuristike za ovaj odabir, no njihova je implementacija znatno složenija, a kako je dani problem relativno uspješno riješen i jednostavnijom heuristikom, složenije heuristike nisu ni razmatrane u sklopu ovog rada.

Kao što je i uobičajeno ([7]), implementirano je spremanje trenutnih odstupanja od tražene vrijednosti u pričuvnu memoriju (eng. *Error Cache*). Spremanje odstupanja u pričuvnu memoriju dovodi do značajnog ubrzanja u postupku rješavanja optimizacijskog problema Stroja s potpornim vektorima, odnosno skraćuje vrijeme učenja.

Ostvareno je i pamćenje vrijednosti jezgrene funkcije za parove primjera za učenje $x^{(i)}, x^{(j)}$ u pričuvnoj memoriji (eng. *Kernel Cache*). Ovim se postupkom postiže značajno ubrzanje jer se prilikom učenja algoritma za sve parove primjera za učenje vrijednosti jezgre izračunavaju samo jednom, a ne u svakom koraku. Pamćenje vrijednosti jezgre u pričuvnoj memoriji moguće je ostvariti jer skup za učenje broji dovoljno malen broj primjera za učenje.

Nad ulaznim vektorima značajki provedena je normalizacija na interval $<0,1>$ da bi Stroj s potpornim vektorima radio ispravno s bilo kojom od realiziranih jezgri. Osim mogućnosti ne korištenja jezgre, kada je Stroj s potpornim vektorima praktički identičan Klasifikatoru optimalne granice, implementirane su i jezgre: Polinomna jezgra, Gaussova jezgra i Racionalna kvadratna jezgra.

3.2.2. Vizualizator Stroja s potpornim vektorima

Za potrebe programskog ostvarenja Stroja s potpornim vektorima izgrađen je i vizualizator.

Vizualizator je korišten za provjeru ispravnosti ostvarenog Stroja s potpornim vektorima, te za vizualizaciju djelovanja pojedinih jezgara na mogućnosti klasifikatora.

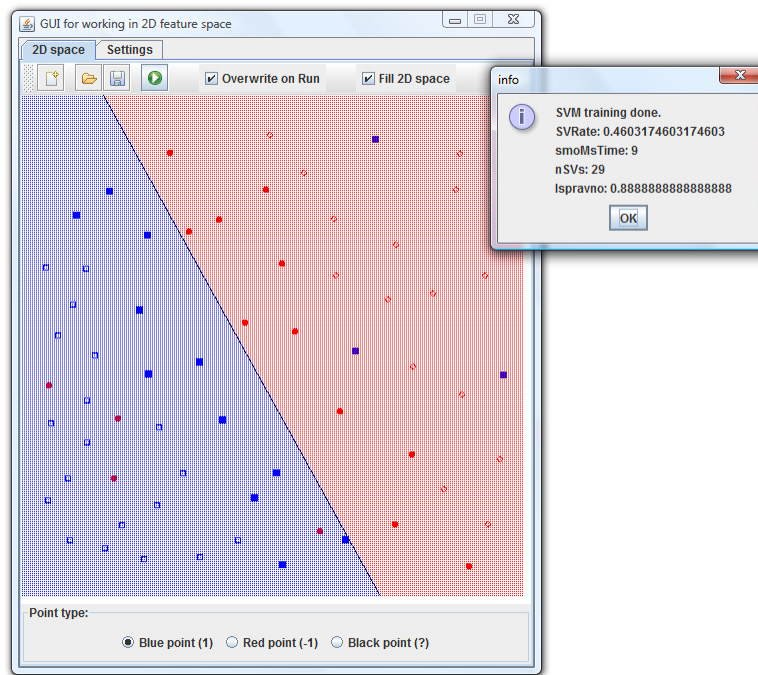
Vizualizator omogućava jednostavno stvaranje skupa za učenje gdje svaki primjer za učenje ima vektor značajki od dva elementa. Kako se vektor značajki sastoji od dva elementa, moguće je svaki primjer za učenje prikazati kao točku u 2D prostoru, gdje je po jedan element na svakoj koordinatnoj osi. Na ovaj je način moguće vizualizirati rezultatnu podjelu prostora na pojedine grupe.

Stvoreni skup za učenje sastoji se od proizvoljnog broja točaka (primjera za učenje) koje pripadaju grupama s oznakom „1“ ili „-1“. Točke koje pripadaju grupi s oznakom „1“ prikazane su u plavoj boji, te su iscrtane u obliku kvadrata, a točke grupe s oznakom „-1“ su prikazane u crvenoj boji u obliku kruga.

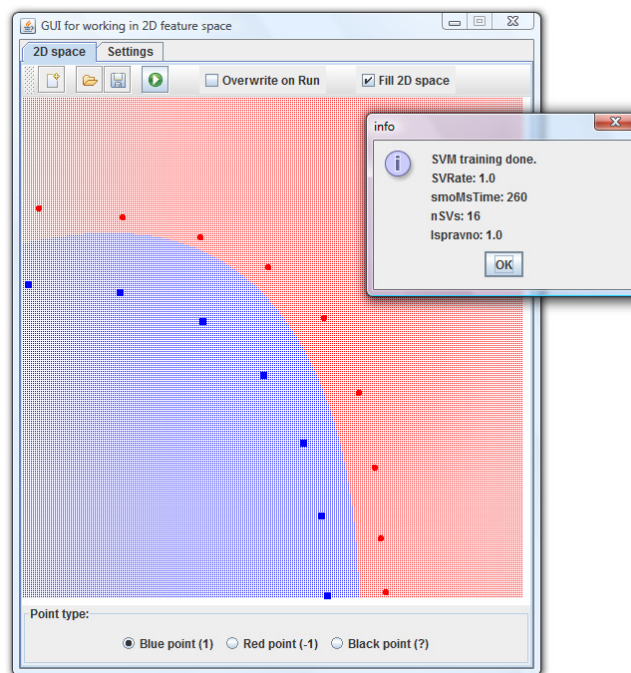
Nakon stvaranja klasifikatora, prostor je vidljivo podijeljen na dvije grupe, tako da je djelom obojan plavom, a djelom crvenom bojom.

Točke koje su potporni vektori, ispunjene su bojom, pa ih je moguće razlikovati od običnih točaka.

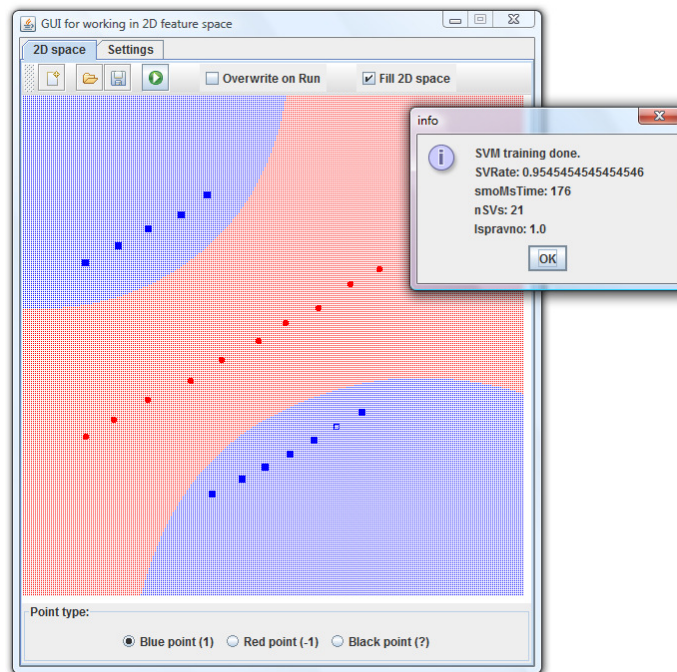
U nastavku su prikazani rezultati Vizualizatora na nekoliko primjera, gdje su upotrijebljivane različite jezgre. Iz prikazanog je vidljivo da izbor jezgre značajno utječe na mogućnost odvajanja različitih grupa. Tako su mogućnosti postignute s Gaussovom jezgrom značajno veće nego one postignute bez upotrebe jezgre. Tokom podešavanja parametara je potrebno paziti da se postigne dobra generalizacija za željeni slučaj. Treba izbjegavati pretjerano prilagođavanje klasifikatora određenom skupu za učenje (eng. *Overfitting*), kao i nedovoljno prilagođavanje klasifikatora danom skupu za učenje (eng. *Underfitting*).



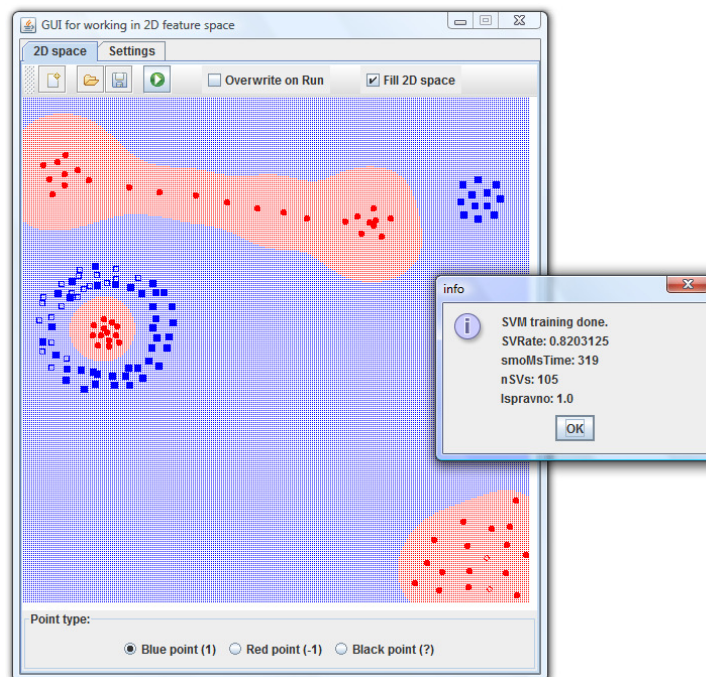
Slika 3.3. Stroj s potpornim vektorima bez jezgre



Slika 3.4. Stroj s potpornim vektorima s Polinomnom jezgrom



Slika 3.5. Stroj s potpornim vektorima s Racionalnom kvadratnom jezgrom



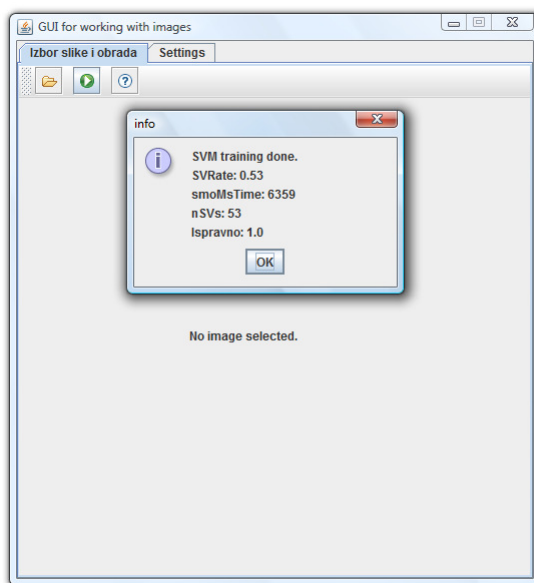
Slika 3.6. Stroj s potpornim vektorima s Gaussovom jezgrom

3.2.3. Sustav za prepoznavanje prometnih znakova

Ostvaren je sustav pomoću kojeg je moguće primijeniti Stroj s potpornim vektorima na klasifikaciju prometnih znakova.

Razvijeni sustav dijeli veći dio mogućnosti s Vizualizatorom, te se koristi na sličan način. Glavna je razlika u nemogućnosti kreiranja vlastitih skupova za učenje, već u korištenju postojećih. Program učitava skup za učenje iz odabranog direktorija u kojem se nalaze poddirektoriji za klasu „-1“ (direktorij „c-1“) i klasu „1“ (direktorij „c1“). Primjeri za učenje su predstavljeni pripadnim normaliziranim fotografijama. Na temelju normaliziranih fotografija generiraju se vektori značajki. Pokretanjem učenja prvo se izvrši spremanje svih vrijednosti jezgrene funkcije u pričuvnu memoriju, a zatim se vrši sam postupak učenja. Nakon što je učenje gotovo (tj. izgrađen je klasifikator) dobije se izvješće o rezultatima učenja, točnije, podatci o utrošenom vremenu, o udjelu i broju potpornih vektora u cijelom skupu za učenje, te podatci o uspješnosti učenja na danom skupu za učenje (udjel ispravno klasificiranih primjera za učenje).

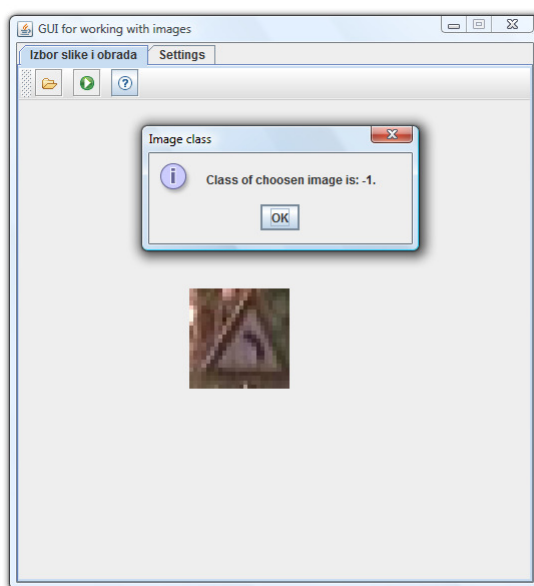
Kad je učenje gotovo, moguće je primijeniti izgrađeni klasifikator za klasifikaciju pojedinih znakova. Klasifikacija se vrši odabirom normalizirane fotografije znaka, a klasifikator vrati oznaku grupe kojoj ispitivani znak pripada. Prikazana je i uvećana ispitivana fotografija.



Slika 3.7 Rezultati učenja klasifikatora za raspoznavanje prometnih znakova upozorenja o zavoju u lijevo i upozorenja o zavoju u desno

Slika 3.7 prikazuje rezultate učenja klasifikatora na prometnim znakovima upozorenja o zavoju u lijevo i upozorenja o zavoju u desno. Vidi se da je udjel potpornih vektora 53% skupa za učenje. Ovo je relativno visok postotak koji indicira da je skup za učenje nedovoljno velik i da bi trebalo dodati još primjera za učenje. Vidljiva je i uspješnost učenja od 100%.

Slika 3.8 prikazuje rezultate klasifikacije jednog prometnog znaka upozorenja o zavoju u lijevo. Znak je ispravno klasificiran jer ima oznaku grupe „-1“, a u skupu za učenje su znakovi upozorenja o zavoju u lijevo zbilja imali oznaku „-1“.



Slika 3.8. Rezultat klasifikacije prometnog znaka

3.3. Rezultati

Provedena je klasifikacija znakova na različitim trokutastim prometnim znakovima. Postignuti su rezultati klasifikacije prometnih znakova metodom potpornih vektora od 90% - 95%. Problem s pouzdanošću ovog iznosa leži u činjenici da je upotrijebljeni skup za učenje bio značajno manji od optimalnog za ovakav klasifikacijski zadatak. Pouzdanost varira ovisno o izboru grupa prometnih znakova koje se suprotstavljaju. Ako se izaberu grupe sličnih znakovi, pouzdanost je lošija, nego ako se izaberu grupe znakova koji se više međusobno razlikuju.

Na raspolaganju je skup od šestotinjak znakova. Ovo bi bio skup za učenje skoro primjerene veličine da se sastoji samo od dvije grupe znakova. No ovaj se skup sastoji od devet grupa znakova. Tako je u svakoj grupi znakova samo pedesetak znakova. To nikako nije dovoljan broj uzme li se u obzir da se vektor značajki minimalno (bez primjene jezgre) sastoji od skoro dvije tisuće elemenata. To znači da se gradi klasifikator u prostoru od skoro dvije tisuće dimenzija, a na raspolaganju je samo stotinjak primjera za učenje kojima se određuje odvajajuća hiperravnina.

Moguće je kombinirati više grupa znakova u jednu, pa klasificirati znak u jednu od dvije veće grupe. Tada se postiže ispravna klasifikacija od oko 80%. Postotak varira u ovisnosti o načinu grupiranja manjih grupa znakova u veće. Pad u pouzdanosti klasifikacije je razumljiv s obzirom da se spajanjem grupa različitih znakova u jednu gubi uniformnost objekata u grupi, što nužno uzrokuje pad pouzdanost klasifikacije jer se generalizacija prilikom učenja ne može u potpunosti postići unutar grupe. O nepouzdanosti ovakve klasifikacije dovoljno govori činjenica da se podjelom svih dostupnih vrsta znakova u dvije skupine, gdje su u jednoj četiri vrste prometnih znakova, a u drugoj pet vrsta prometnih znakova, uspijeva učenjem dobiti uspješnost od 98.7%, a da je na testnom skupu dobije pouzdanost od 75%. Ako se svi znakovi testnog skupa pravilno rasporede u pripadne grupe, i dalje se postiže uspješnost od 98.4%. Ovakvo ponašanje potvrđuje činjenicu da je glavni problem dane klasifikacije nedostatna veličina skupa za učenje.

Iznesheni rezultati se odnose na primjenu Stroja s potpornim vektorima bez jezgre. Rezultati Stroja s potpornim vektorima s jezgrom su praktički identični, što je razumljivo jer su grupe znakova razdvojive u prostoru vektora značajki bez upotrebe jezgre, te su zato očekivano razdvojive i u prostoru s većim brojem dimenzija.

Koristi li se Stroj s potpornim vektorima na ovakvim malim skupovima za učenje, a s značajno većim vektorima značajki, razumnije je koristiti klasifikator bez jezgre jer se postižu slični rezultati kao i s upotrebom jezgre, a nema problema s odabirom odgovarajućih (optimalnih) parametara jezgre.

3.4. Moguća poboljšanja

Performanse razvijenog sustava za klasifikaciju prometnih znakova pomoću Stroja s potpornim vektorima moguće je značajno poboljšati upotrebom niza postupaka. Glavni i

nužni preduvjet svim poboljšanjima je dostupnost većih grupa prometnih znakova da bi skupovi za učenje bili dovoljno veliki, a time i provedena generalizacija dovoljno robusna.

Skoro je sigurno moguć bolji izbor vektora značajki. Da bi se izabrao što bolji skup značajki potrebno je testirati performanse sustava upotrebom različitih skupova značajki. Neki primjeri alternativnih skupova značajki: moguće je značajke odabrati u drugom sustavu boja, moguće je značajke temeljiti na pojavama određenih uzoraka u prometnom znaku, moguće je kao značajke uzimati nagibe pravaca koji aproksimiraju detektirane rubove u prometnom znaku... Moguće je koristiti i kombinaciju nekoliko prethodno navedenih skupina značajki.

Moguće je specijalizirati sustav više prema klasifikaciji fotografija. Tako je moguće pretprocesiranjem popraviti ulazne fotografije, npr. odgovarajućim filterima ukloniti ili smanjiti iskrivljenost fotografija koja nastaje kao posljedica gibanja kamere tokom fotografiranja. Postoje i određene jezgre koji su kreirane specifično za problem klasifikacije slika ([10]). Moguće je čak i kreirati specifičnu jezgru baš za problem klasifikacije prometnih znakova.

Konačno, moguće je značajno smanjiti vrijeme potrebno da se klasifikator izgradi. Istraživanjima su do sada otkrivene razne metode ubrzavanja algoritma Slijedne minimalne optimizacije ([11]), od izbora bolje heuristike, do upotrebe složenih statističkih metoda. Vjerojatno je moguće je i sam algoritam bolje ostvariti u izvornom kodu.

Da bi implementacija bilo kojeg od navedenih postupaka bila opravdana i izvediva, nužna je dostupnost većeg skupa za učenje.

Što se tiče samog programskog ostvarenja, razumno bi bilo koristiti već gotovu implementaciju Stroja s potpornim vektorima. Ovakav izbor ima cijeli niz prednosti, a mali broj nedostataka. Prednosti uključuju: značajno kraće vrijeme potrebno za učenje, automatiziranu selekciju optimalnih parametara jezgre, veću sigurnost u ispravnost sustava, slobodniji izbor značajki, mogućnost upotrebe više od dvije grupe za klasifikaciju... Jedini pravi nedostaci su teže dodavanje novih jezgara ili modifikaciji postupaka koji se provode. Ovi nedostaci su neznatni nasuprot prednosti, pogotovo jer ih je moguće ispraviti. Postoji nekoliko besplatnih, kvalitetnih sustava ostvarenih u različitim programskim jezicima, pa je moguće i modificirati postojeće sustave. Primjer sustava koji bi bio dobar izbor je LIBSVM sustav ([12]). Ovaj sustav je besplatan i ima sve prethodno navedene prednosti.

Zaključak

Raspoznavanje ili klasifikacija objekta je proces određivanja pripadnosti objekta određenoj grupi (klasi) objekata sa sličnim odabranim svojstvima. Rezultati klasifikacije uvelike ovise o algoritmu kojim se klasifikacija vrši.

Postoji veoma velik broj različitih algoritama za klasifikaciju, a ovdje proučavani Stroj s potpornim vektorima je klasifikacijski algoritam iz skupine algoritama za nadgledano strojno učenje. Potrebno je prirediti skup za učenje na temelju kojeg će se izgraditi klasifikator, a nakon što je klasifikator izgrađen, moguće je njime vršiti neovisne klasifikacije. Skup za učenje se sastoji od unaprijed pripremljenih ispravnih klasifikacija objekata koje su predstavljene kao parovi vektora svojstava (značajki) nekog objekta i pripadne, ispravne klasifikacije tog objekta.

Stroj s potpornim vektorima se temelji na starijem algoritmu Klasifikatora optimalne granice, no nije ograničen na klasifikaciju linearno odvojivih grupa, te dodatno ispravno funkcionira i ako grupe nisu u potpunosti odvojive. Zbog samog načina funkcioniranja algoritma, moguće je prilično jasno analizirati interno ostvarenje stvorenog klasifikatora, što kod nekih drugih algoritama nije moguće (npr. kod Neuronskih mreža).

Stroj s potpornim vektorima je algoritam koji mnogi stručnjaci smatraju najučinkovitijim standardnim (eng. *of-the-shelf*) algoritmom za nadgledano strojno učenje. Iako taj stav ne dijele svi stručnjaci u području strojnog učenja, nitko ne dovodi u pitanje kvalitetu i značaj ovog algoritma.

Izložen je i jedan algoritam kojim je moguće provesti učenje Stroja s potpornim vektorima, kao i moguća poboljšanja performansi tog algoritma u odnosu na implementiranu verziju.

Stroj s potpornim vektorima je primijenjen na problem klasifikacije prometnih znakova. Ostvaren je jedan način reprezentacije prometnih znakova odgovarajućim vektorima značajki, a predložene su dodatne obećavajuće mogućnosti. Izloženi su rezultati primjene izgrađenog klasifikatora na problem prepoznavanja prometnih znakova. Rezultati su zadovoljavajući s obzirom na dane uvjete i ograničenja problema, gdje je glavni ograničavajući faktor nedostatna veličina skupa za učenje.

Kad se sve uzme u obzir, nedvosmisleno je jasno da je primjena Stroja s potpornim vektorima na problem klasifikacije prometnih znakova veoma obećavajuća metoda. Potrebno je osigurati veći skup za učenje, te provesti dodatne analize da bi se dobila maksimalna pouzdanost i učinkovitost ove metode.

Literatura

- [1] Burges, Christopher J. C.: A Tutorial on Support Vector Machines for Pattern Recognition, *Data Mining and Knowledge Discovery*, vol. 2, pp. 121-164, 1998.
- [2] Boser et al: A Training Algorithm for Optimal Margin Classifiers, *Proceedings Fifth ACM Workshop on Computational Learning Theory*, pp. 144–152, 1992.
- [3] Shawe-Taylor, John; Cristianini, Nello: *Kernel Methods for Pattern Analysis*, Cambridge University Press, New York, USA, 2004.
- [4] Genton, Marc G.: Classes of kernels for machine learning: a statistics perspective, *Journal of Machine Learning Research*, vol. 2, pp. 299-312, 2002.
- [5] Wu, Q.; Zhou, D.: SVM Soft Margin Classifiers: Linear Programming versus Quadratic Programming. *Neural Computation*, vol. 17, pp. 1160-1187, 2005.
- [6] Hush, Don; Scovel, Clint: Polynomial-time decomposition algorithms for support vector machines, *Machine Learning*, vol. 51, pp. 51–71, 2003.
- [7] Platt, John C.: Fast Training of Support Vector Machines using Sequential Minimal Optimization, u knjizi Schölkopf et al: *Advances in Kernel Methods - Support Vector Learning*, Cambridge, MA, 1998. MIT Pres
- [8] Lin, Chih-Jen: Asymptotic Convergence of an SMO Algorithm Without Any Assumptions, *IEEE Transactions on Neural Networks*, vol. 13, pp. 248–250, 2002.
- [9] O. Chapelle, P. Haffner, and V. Vapnik, "SVMs for histogram-based image classification," *IEEE Transactions on Neural Networks*, vol. 9, 1999.
- [10] Zamolotskikh, A; Cunningham, P.: An Assessment of Alternative Strategies for Constructing EMD-Based Kernel Functions for Use in an SVM for Image Classification, *CBMI '07 International Workshop*, pp. 11-17, 2007.
- [11] Keerthi et al: Improvements to Platt's SMO algorithm for SVM classifier design, *Neural Computation*, vol. 13, pp. 637–649, 2001.
- [12] Chang, Chih-Chung; Lin, Chih-Jen: LIBSVM: a library for support vector machines, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

Naslov, sažetak i ključne riječi

Naslov:

Raspoznavanje prometnih znakova metodom potpornih vektora

Sažetak:

Prezentiran je klasifikacijski algoritam Stroja s potpornim vektorima. Definirana su željena svojstva klasifikatora. Vodeći se ciljanim svojstvima, prvo je matematički formuliran optimizacijski problem Klasifikatora optimalne margine, a potom prezentiran algoritam Slijedne minimalne optimizacija koji rješava zadani optimizacijski problem. Na Klasifikator optimalne margine je primijenjen jezgreni trik, te je tako dobiven Stroj s potpornim vektorima. Objašnjene su neke često korištene jezgre poput Gaussove, Polinomne i Racionalne kvadratne jezgre. Potom je predstavljen problem raspoznavanja prometnih znakova. Opisani su ulazni podaci, njihova obrada, te priprema za Stroj s potpornim vektorima. Prezentirani su rezultati, kao i razne mogućnosti poboljšanja performansi opisanog i razvijenog sustava.

Ključne riječi:

Stroj s potpornim vektorima, metoda potpornih vektora, raspoznavanje objekata, raspoznavanje prometnih znakova

Title, abstract and keywords

Title:

Traffic sign classification using Support Vector Machines

Abstract:

Description of classification algorithm called Support Vector Machine (SVM) is given. Desired properties for classifier are defined. Keeping that properties in mind, mathematical formulation of optimization problem for Optimal Margin Classifier is given and then solved using Sequential Minimal Optimization (SMO). Method called Kernel Trick is described and applied to Optimal Margin Classifier resulting in Support Vector Machine algorithm. Various general purpose kernels, such as Gaussian, Polynomial and Rational quadratic kernel are presented. Finally, traffic signs classification problem is introduced. Input data, input data processing and preparation for use with SVMs are described. Achieved results are presented, along with some methods which could improve performance of described and developed system.

Keywords:

Support Vector Machine, SVM, object classification, traffic signs classification