Pronalaženje skrivenog znanja

Elektrotehnički fakultet Univerziteta u Beogradu

Master akademske studije

modul Računarska tehnika i informatika

2017/2018

Metoda potpornih vektora

Vuk Batanović, Elektrotehnički fakultet Univerziteta u Beogradu

Metoda potpornih vektora

- Metoda potpornih vektora (engl. *Support Vector Machines – SVM*)
- Binarni klasifikator
  - Najčešće se jedna klasa označava sa $y = +1$ a druga sa $y = -1$
- Osnovna verzija algoritma je linearni klasifikator
- Neprobabilistički klasifikator – izlaz modela je samo klasifikaciona odluka
- Pošto direktno modeluje granicu između klasa, metoda potpornih vektora spada u diskriminativne modele

Metoda potpornih vektora – istorijat

- Osnovnu varijantu metode potpornih vektora su predstavili Vladimir Vapnik i Aleksej Červonenkis 1963. godine
- 1992. godine je grupa autora (uključujući Vapnika) predložila proširenje na nelinearne slučajeve korišćenjem kernelskog trika (engl. *kernel trick*)
- Kortez i Vapnik su 1993. godine predložili, a 1995. godine objavili varijantu algoritma sa mekim marginama (engl. *soft margin*)

Izbor hiperravni razdvajanja

- Pri klasifikaciji podataka potrebno je pronaći hiperravan koja razdvaja podatke koji pripadaju različitim klasama
- Ako su podaci linearno separabilni moguće je pronaći beskonačno mnogo hiperravni koje su u stanju da izvrše razdvajanje podataka iz skupa za obučavanje
- Pitanje – po kom kriterijumu odabrati jednu od njih?
  - Logistička regresija – kriterijum je maksimalna entropija modela
  - Metoda potpornih vektora – kriterijum je maksimalna margina između klasa

Maksimalna margina razdvajanja i potporni vektori

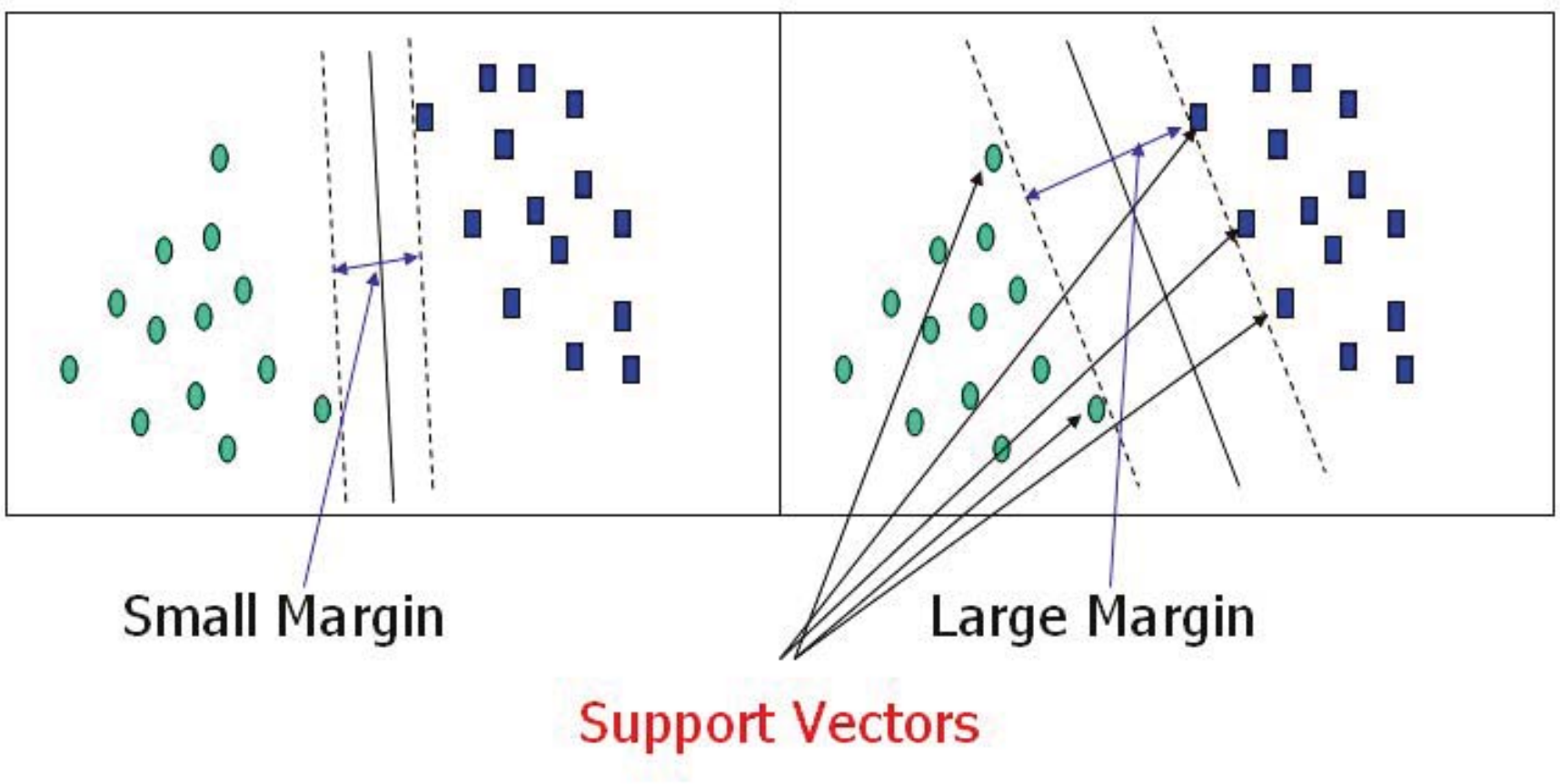
- Ako su podaci linearno separabilni, onda je moguće zamisliti dve paralelne hiperravni takve da razdvajaju podatke različitih klasa i da se između njih ne nalazi ni jedan podatak
- Oblast prostora između ovih zamišljenih hiperravni se naziva *marginom*
- Cilj obučavanja modela u metodi potpornih vektora je da se pronađe maksimalna margina
- Obe zamišljene hiperravni efektivno naležu na jedan ili više podataka svoje klase – ti podaci su *potporni vektori* (engl. *support vectors*)
  - Potporni vektori „podupiru“ zamišljene hiperravni
  - Naziv *vektor* potiče od posmatranja svakog podatka kao vektora/tačke u n-dimenzionalnom prostoru

Maksimalna margina razdvajanja i potporni vektori

- Intuitivno gledano, hiperravan razdvajanja koja prolazi kroz sredinu praznine između podataka dveju klasa ima više smisla od one koja se nalazi blizu podataka bilo jedne bilo druge klase
- Biranjem hiperravni razdvajanja koja se nalazi najdalje od uočenih podataka efektivno se minimizuje greška generalizacije modela (tj. greška na novim podacima)
  - Pod pretpostavkom da će i novi podaci biti generisani iz iste raspodele kao i postojeći
- Ovo se razlikuje od pristupa u logističkoj regresiji gde se minimizuje empirijska greška na već raspoloživim podacima za obučavanje (tj. maksimizuju performanse modela na njima)

Maksimalna margina razdvajanja i potporni vektori

- Pri traženju hiperravni razdvajanja sa maksimalnom marginom položaj podataka koji su udaljeniji od hiperravni razdvajanja nije bitan – oni ne utiču na marginu razdvajanja podataka
- Hiperravan razdvajanja je određena malim brojem podataka obe klase koji su joj najbliži
  - Oni su najvažniji za pravilnu klasifikaciju kako postojećih tako i novih podataka
- Ovo je drugačije od logističke regresije gde hiperravan razdvajanja zavisi od položaja svih podataka



Ilustracija veličine margine razdvajanja i potpornih vektora

Slika preuzeta sa: http://www.dtreg.com/solution/view/20

Hipoteza metode potpornih vektora

- U metodi potpornih vektora model predstavlja jednačinu hiperravni razdvajanja
- Ako je n broj odlika koje se koriste u modelu tj. broj dimenzija prostora, onda je hipoteza oblika:

$$h(x) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n = W^T X + w_0$$

- Često se umesto oznake w_0 koristi oznaka b (engl. *bias term*)
- Jednačina hiperravni razdvajanja je:

$$h(x) = W^T X + w_0 = 0$$

- Vektor W je vektor normale na hiperravan razdvajanja

Skaliranje jednačine hiperravni razdvajanja

- Jednačina hiperravni razdvajanja se može pomnožiti/podeliti bilo kojim pozitivnim brojem bez promene položaja hiperravni
- Iako je faktore W i w_0 moguće proizvoljno skalirati, konvencionalno se usvaja takva kanonska vrednost da za potporne vektore $X^{(sv)}$ važi

$$y^{(sv)}\left(W^T X^{(sv)} + w_0\right) = 1$$

- To znači da jednačina hiperravni koja „naleže“ na potporne vektore iz klase $y = 1$ postaje:

$$W^T X + w_0 = 1$$

- dok je jednačina hiperravni koja „naleže“ na potporne vektore iz klase $y = -1$:

$$W^T X + w_0 = -1$$

Širina margine razdvajanja

- Udaljenost neke tačke/podatka $X^{(a)}$ od hiperravni razdvajanja Π je:

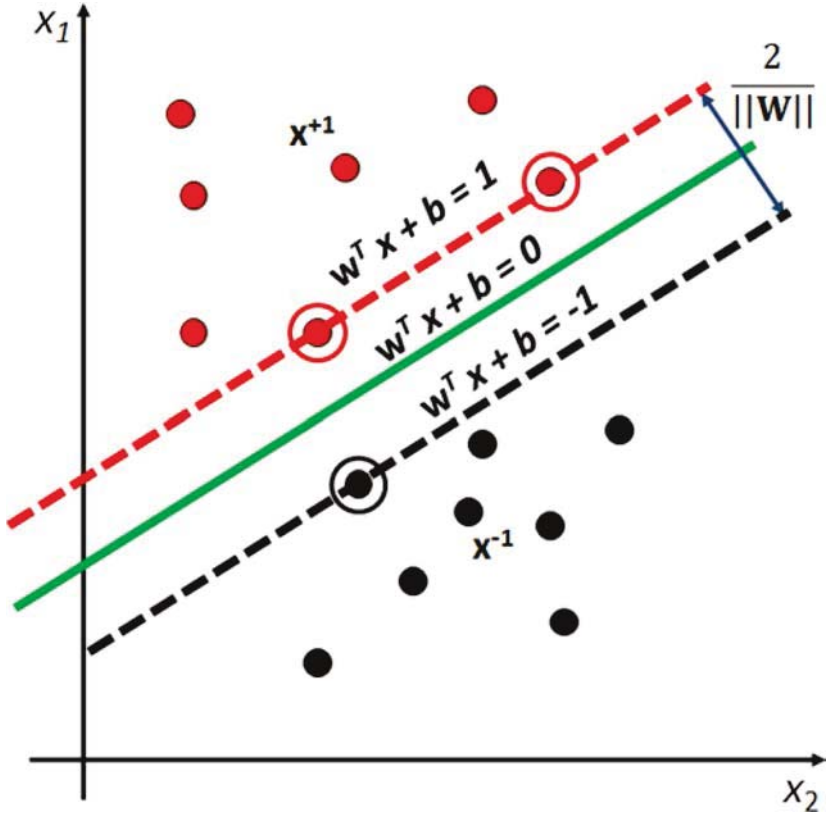
$$dist\left(X^{(a)}, \Pi\right) = \frac{\left|W^T X^{(a)} + w_0\right|}{\|W\|_2}$$

- Sledi da je udaljenost potpornih vektora $X^{(sv)}$ od hiperravni razdvajanja:

$$dist\left(X^{(sv)}, \Pi\right) = \frac{\left|W^T X^{(sv)} + w_0\right|}{\|W\|_2} = \frac{1}{\|W\|_2}$$

- Pošto se potporni vektori nalaze sa obe strane hiperravni razdvajanja, širina margine je dvostruko veća – nju treba maksimizovati:

$$argmax_{W,w_0} \frac{2}{\|W\|_2}$$



Udaljenost potpornih vektora od hiperravni razdvajanja

Slika preuzeta sa: http://jap.physiology.org/content/115/11/1714

Cilj obučavanja u metodi potpornih vektora

- Pošto marginu treba maksimizovati, sledi da treba minimizovati $\|W\|_2$
- Pošto je

$$\|W\|_2 = \sqrt[2]{w_1^2 + w_2^2 + \cdots + w_n^2}$$

- sa stanovišta optimizacije lakše je izbeći korenu funkciju i minimizovati $\frac{1}{2}\|W\|_2^2$
  - Ovu zamenu je moguće sprovesti jer minimizovanje kvadrata norme minimizuje i samu normu
- Stoga je cilj obučavanja u metodi potpornih vektora pronaći

$$argmin_{W,w_0} \frac{1}{2}\|W\|_2^2$$

Uslovi pri traženju maksimalne margine

- Prilikom pronalaženja maksimalne margine treba sprečiti da neki podatak bilo prve bilo druge klase upadne u marginu, što znači da treba da važi:

$$W^T X^{(+1)} + w_0 \geq 1$$

- gde su $X^{(+1)}$ podaci klase $y = +1$, odnosno:

$$W^T X^{(-1)} + w_0 \leq -1$$

- gde su $X^{(-1)}$ podaci klase $y = -1$
- Navedeni uslovi se mogu uprostiti na:

$$y^{(i)}(W^T X^{(i)} + w_0) \geq 1$$

Optimizacioni problem

- Optimizacioni problem ima sledeći oblik:

$$argmin_{W,w_0} \frac{1}{2}\|W\|_2^2, \qquad y^{(i)}\left(W^T X^{(i)} + w_0\right) \geq 1, \qquad i = 1,2,\dots,m$$

- gde je m broj podataka u skupu za obučavanje
- Ovo je problem konveksne optimizacije sa ograničenjem, tj. kvadratnog programiranja
- Do rešenja se može doći tehnikom Lagranžovih multiplikatora uz *Karush-Kuhn-Tucker*-ove (KKT) uslove
- Ovo rešenje podrazumeva kreiranje tzv. dualne reprezentacije problema u kojoj se za svaku nejednakost navedenog oblika kreira po jedan Lagranžov multiplikator α_k

Dualna formulacija hipoteze metode potpornih vektora

- U dualnoj reprezentaciji problema dobija se da je:

$$W = \sum_{i=1}^{m} \alpha_i y^{(i)} X^{(i)}$$

$$h(x) = W^T X + w_0 = \sum_{i=1}^{m} \alpha_i y^{(i)} \left(X^{(i)}\right)^T X + w_0$$

- Pokazuje se da su optimalne vrednosti α_i dodeljene svakoj tački jednake nuli za sve tačke osim za potporne vektore
  - Njih obično ima mnogo manje od ukupnog broja tačaka
- Stoga vrednost hipoteze (i jednačina hiperravni razdvajanja) zavise samo od potpornih vektora

Dualna formulacija hipoteze metode potpornih vektora

- Vrednost hipoteze $h\left(x^{(k)}\right)$ za neki nov podatak $x^{(k)}$ zavisi od proizvoda

$$\left(X^{(i)}\right)^T X^{(k)} = X^{(i)} \cdot X^{(k)}$$

- tj. od sličnosti potpornih vektora $X^{(i)}$ sa vektorom novog podatka $X^{(k)}$
- Klasifikovanje podatka predstavlja proveru sa koje strane hiperravni razdvajanja se on nalazi:

$$\hat{y} = sign(h(x))$$

Meka margina

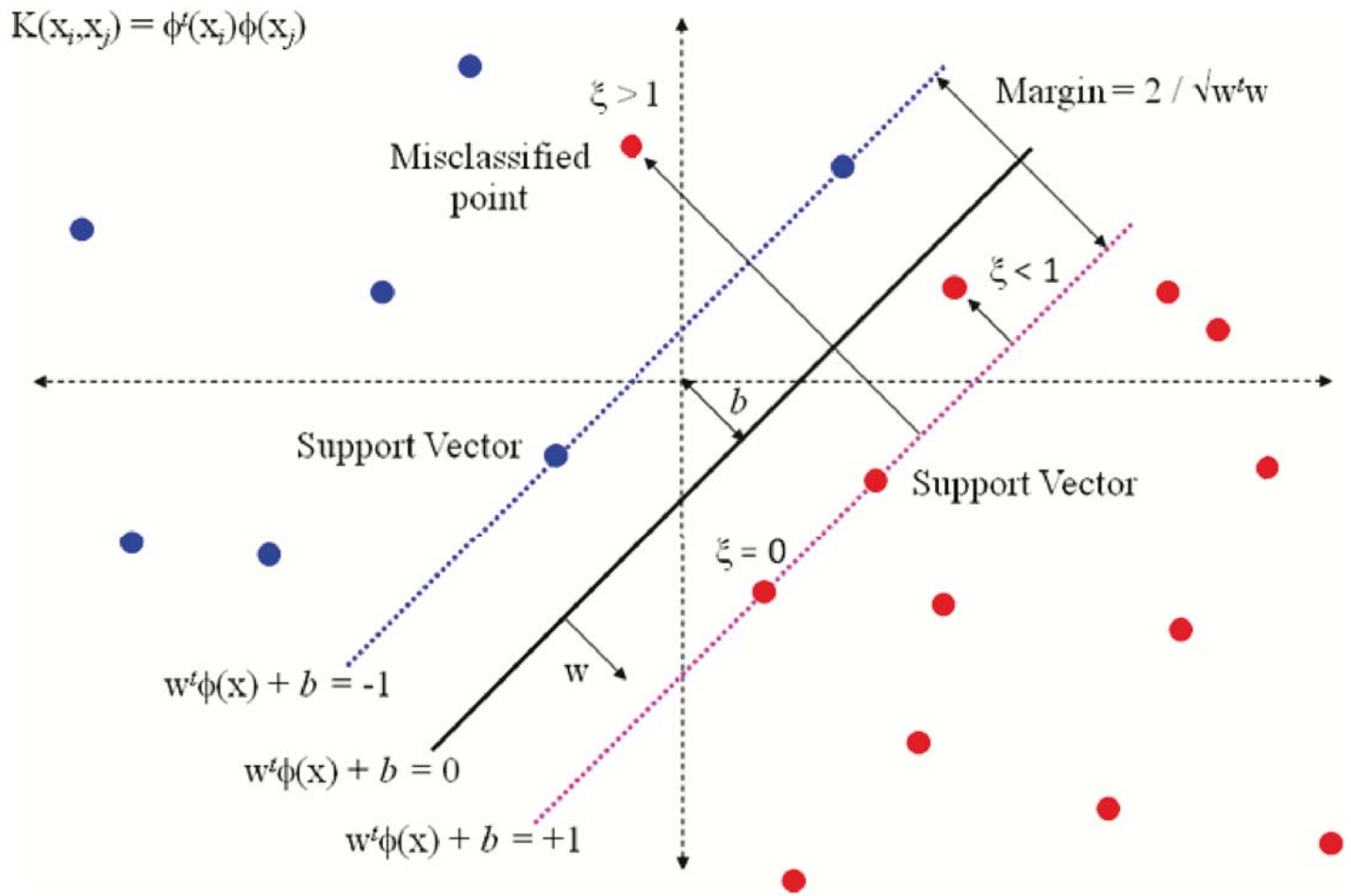
- Insistiranje na linearnoj separabilnosti klasa, čak i ako je ona moguća, može da dovede do preterane prilagođenosti modela podacima
  - Pravi probleme u slučaju prisustva *outlier*-a – pronađeno rešenje će imati malu marginu tj. slabu generalizaciju na nove podatke
- Umesto toga, bolje je (u ograničenoj meri) dozvoliti ulaske podataka u marginu ili njihov prelazak na pogrešnu stranu hiperravni razdvajanja
- Margina u kojoj je ovo dozvoljeno se naziva mekom marginom (engl. *soft margin*)

Meka margina

- Korišćenje meke margine dovodi do reformulacije optimizacionog problema:

$$argmin_{W,w_0} \frac{1}{2}\|W\|_2^2, \qquad y^{(i)}\left(W^T X^{(i)} + w_0\right) \geq 1 - \xi_i, \qquad \xi_i \geq 0,$$
$$i = 1,2,\dots,m$$

- gde su ξ_i prekoračenja u položaju svake tačke izvan granice njene klase (engl. *slack variables*)
- Ukoliko je
  - $\xi_i = 0$ – podatak je pravilno klasifikovan i izvan margine
  - $1 \geq \xi_i > 0$ – podatak je pravilno klasifikovan ali unutar margine
  - $\xi_i > 1$ – podatak nije pravilno klasifikovan – nalazi se sa suprotne strane hiperravni razdvajanja u odnosu na njegovu klasu



Ilustracija meke margine u metodi potpornih vektora

Slika preuzeta sa: http://steve-cronin.blogspot.rs/2010/09/modern-analytics-look-at-smo.html

Funkcija gubitka kod SVM-a sa mekom marginom

- Svaki podatak koji uđe u marginu ili pređe na pogrešnu stranu hiperravni razdvajanja treba penalizovati proporcionalno njegovoj udaljenosti od ivice margine na strani klase kojoj pripada
- Funkcija gubitka može imati dva oblika
  - L_1 oblik funkcije gubitka:

$$L(h(x), y) = \xi$$

  - L_2 oblik funkcije gubitka:

$$L(h(x), y) = \xi^2$$

Gubitak u obliku šarke

- Funkcija gubitka SVM-a sa mekom marginom se naziva i gubitkom u obliku šarke (engl. *hinge loss*)
- Uslov meke margine za jedan podatak se može preformulisati kao:

$$y(W^T X + w_0) \geq 1 - \xi$$

$$yh(x) \geq 1 - \xi$$

$$\xi \geq 1 - yh(x)$$

- Ako se uzme u obzir i uslov da je $\xi \geq 0$, sledi:

$$\xi = max\big(0, 1 - yh(x)\big)$$

- Ako se usvoji da je $z = y(W^T X + w_0)$, onda je gubitak u obliku šarke:

$$\xi = max(0, 1 - z)$$

Reformulacija logističke funkcije gubitka za $y \in \{-1,1\}$

- Logistička funkcija gubitka ima sledeći oblik za $y \in \{0,1\}$:

$$L(h(x), y) = -\ln\big(P(y|x)\big) = \begin{cases} -\ln(P(y=1|x)) = -\ln\big(h(x)\big), & y = 1 \\ -\ln(P(y=0|x)) = -\ln\big(1-h(x)\big), & y = 0 \end{cases}$$

- Ako se koriste oznake $y \in \{-1,1\}$ za klase, onda važi:

$$P(y=1|x) = \frac{1}{1+e^{-(W^T X + w_0)}}$$

$$P(y=-1|x) = \frac{1}{1+e^{W^T X + w_0}}$$

$$P(y|x) = \frac{1}{1+e^{-y(W^T X + w_0)}}$$

Reformulacija logističke funkcije gubitka za $y \in \{-1,1\}$

- Sledi da je za $y \in \{-1,1\}$:

$$-\ln\big(P(y|x)\big) = -\ln\frac{1}{1+e^{-y(W^TX+w_0)}} = \ln\left(1+e^{-y(W^TX+w_0)}\right)$$

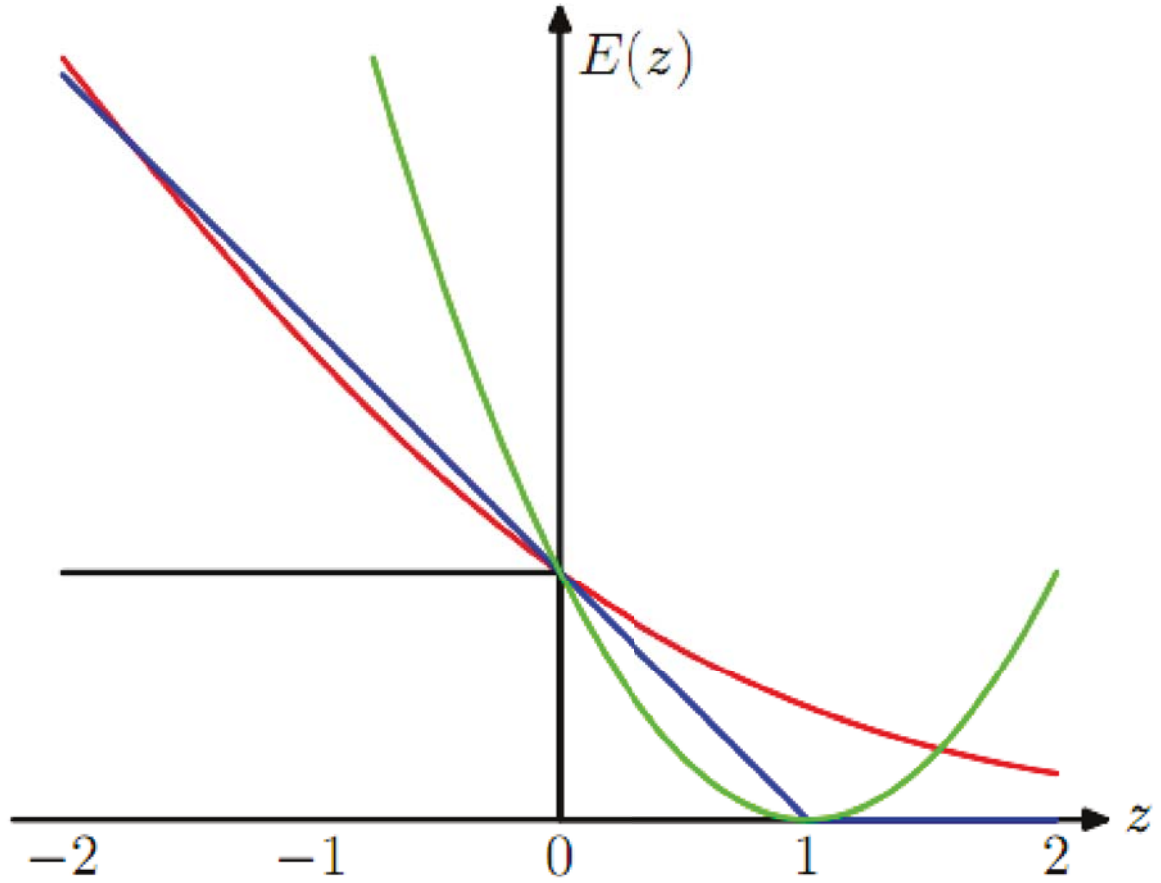
- Logistička funkcija gubitka za $y \in \{-1,1\}$ se obično skalira sa $\frac{1}{\ln 2}$:

$$L(h(x),y) = \frac{1}{\ln 2}\ln\left(1+e^{-y(W^TX+w_0)}\right)$$

- Razlog za skaliranje – ako se usvoji da je $z = y(W^TX + w_0)$, onda logistička funkcija gubitka

$$L(z) = \frac{1}{\ln 2}\ln(1+e^{-z})$$

- prolazi kroz tačku $(0,1)$



Poređenje oblika različitih funkcija gubitka

Legenda: 0-1 gubitak (crna), gubitak u obliku šarke (plava), logistička funkcija gubitka (crvena), kvadratna funkcija gubitka (zelena)

Slika preuzeta iz: Christopher Bishop, Pattern Recognition and Machine Learning

Funkcija greške kod SVM-a sa mekom marginom

- U metodi potpornih vektora sa mekom marginom postoje dva različita (i ponekad suprotstavljena) cilja optimizacije
  - Maksimizacija margine
  - Minimizacija greške klasifikacije na pojedinačnim primerima tj. gubitka u obliku šarke
- Funkcija greške kod SVM-a sa mekom marginom ima oblik:

$$J(w) = C\sum_{i=1}^{m} L\left(h\left(x^{(i)}\right), y^{(i)}\right) + \frac{1}{2}\sum_{j=1}^{n} w_j^2$$

- C je hiperparametar modela koji određuje balans između dva cilja optimizacije
- Cilj obučavanja je minimizacija funkcije greške $J(w)$

Funkcija greške kod SVM-a sa mekom marginom

- Ukoliko se koristi L_1 oblik funkcije gubitka, onda je funkcija greške:

$$J(w) = C\sum_{i=1}^{m}\xi_i + \frac{1}{2}\sum_{j=1}^{n}w_j^2$$

$$= C\sum_{i=1}^{m} max\left(0, 1 - y^{(i)}h\left(x^{(i)}\right)\right) + \frac{1}{2}\sum_{j=1}^{n}w_j^2$$

- Ukoliko se koristi L_2 oblik funkcije gubitka, onda je funkcija greške:

$$J(w) = C\sum_{i=1}^{m}\xi_i^2 + \frac{1}{2}\sum_{j=1}^{n}w_j^2$$

$$= C\sum_{i=1}^{m}\left(max\left(0, 1 - y^{(i)}h\left(x^{(i)}\right)\right)\right)^2 + \frac{1}{2}\sum_{j=1}^{n}w_j^2$$

Funkcija greške kod SVM-a sa mekom marginom

- L_2 oblik funkcije gubitka
  - Rešenje je stabilnije (male promene u podacima za obučavanje manje utiču na rešenje)
  - Manja otpornost na uticaj *outlier*-a
- L_1 oblik funkcije gubitka
  - Rešenje je manje stabilno (male promene u podacima za obučavanje više utiču na izgled rešenja)
  - Veća otpornost na uticaj *outlier*-a

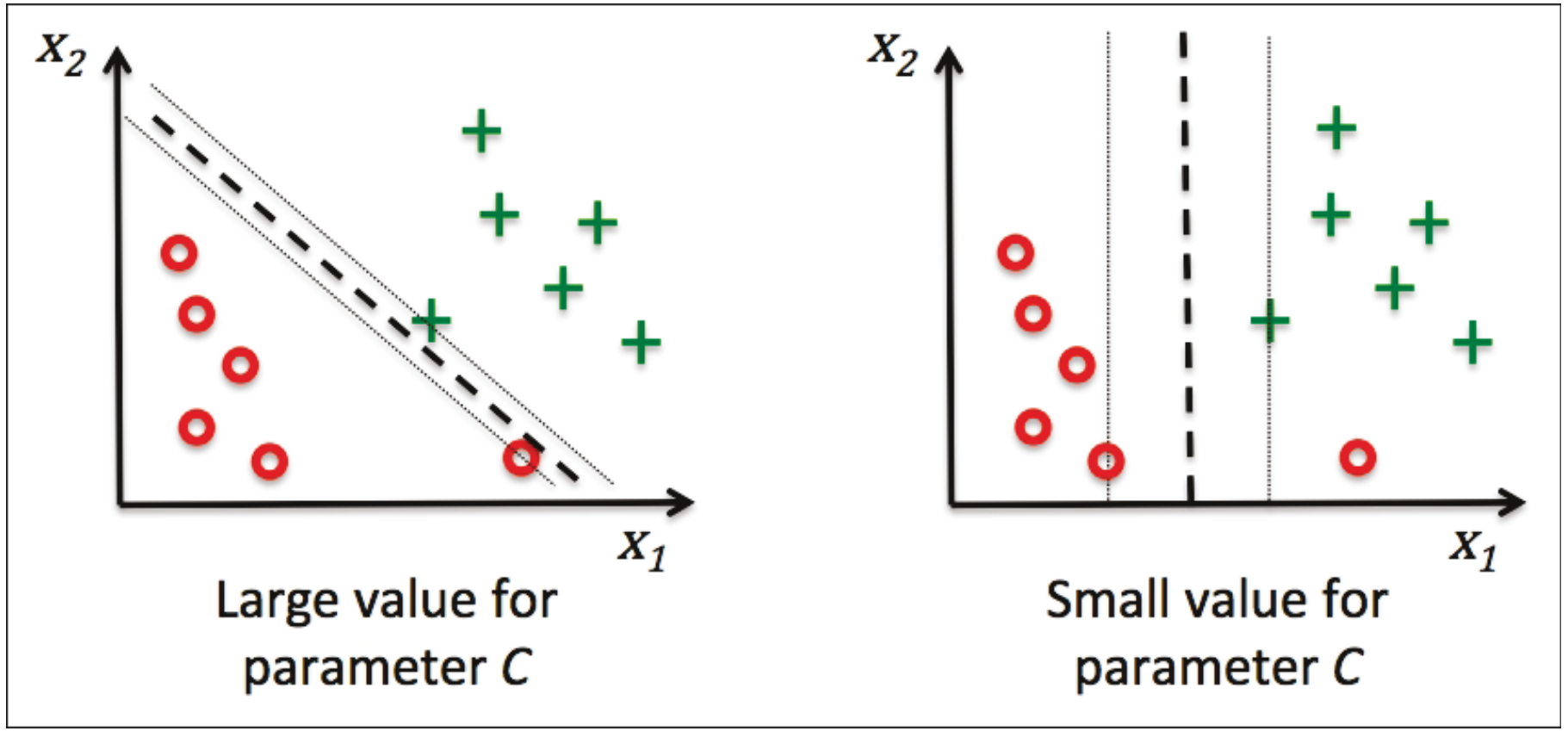
Regularizacija kod SVM sa mekom marginom

- Veća margina povećava generalizaciju modela – minimizovanje norme vektora W predstavlja regularizaciju modela
- Moguće je koristiti L_1 ili L_2 regularizaciju
  - U L_2 regularizaciji se minimuzije do sada korišćen izraz $\frac{1}{2}\|W\|_2^2$
  - U L_1 regularizaciji se minimizuje vrednost $\|W\|_1$
  - Sve ranije prikazane razlike između ova dva tipa regularizacije važe i kod metode potpornih vektora

Hiperparametar C

- Velika vrednost C – model se pri obučavanju fokusira na smanjivanje greške klasifikacije, po cenu manje margine razdvajanja, čime se smanjuje stepen regularizacije
- Mala vrednost C – model se pri obučavanju fokusira na povećanje margine razdvajanja, čime se povećava stepen regularizacije, po cenu veće greške klasifikacije
- Po svojoj funkciji hiperparametar C je inverzan hiperparametru λ koji određuje jačinu regularizacije u linearnoj i logističkoj regresiji:

$$C = \frac{1}{\lambda}$$

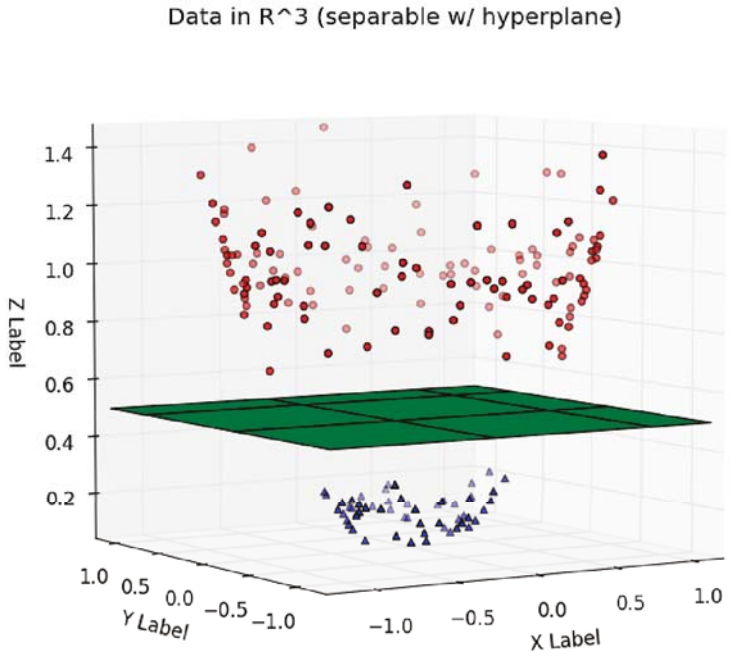


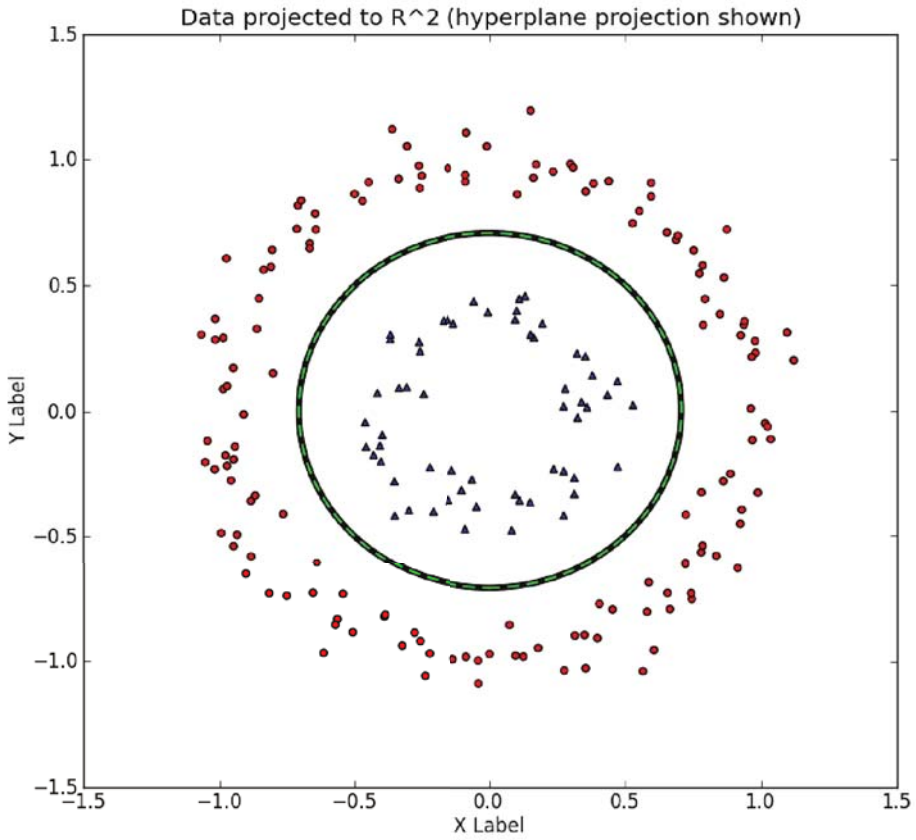
Ilustracija efekta vrednosti hiperparametra C na odabir optimalne hiperravni razdvajanja

Slika preuzeta iz: Sebastian Raschka, Python Machine Learning

Nelinearna klasifikacija

- Metod potpornih vektora se može lako proširiti tako da je u stanju da se nosi i sa klasama podataka koje nisu linearno separabilne
- To se ostvaruje preslikavanjem problema u veći broj dimenzija
  - Ako podaci nisu linearno separabilni u zadatom broju dimenzija, moguće ih je preslikati u prostor sa većim brojem dimenzija u kome će biti moguće linearno ih razdvojiti
  - Linearna granica razdvajanja u prostoru sa više dimenzija, nakon preslikavanja nazad u originalni broj dimenzija, može da bude ekvivalentna sa nelinearnom granicom proizvoljnog oblika
  - Problem – transformacija svih podataka u prostor sa većim brojem dimenzija je računski izuzetno zahtevna operacija





Linearna separacija podataka kroz projekciju u veći broj dimenzija

Slika preuzeta sa: http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

Kernel funkcije

- Hipoteza metode potpornih vektora je sledećeg oblika (u dualnom domenu) za neki podatak $x^{(k)}$:

$$h\left(x^{(k)}\right) = \sum_{i=1}^{m} \alpha_i y^{(i)} \left(X^{(i)}\right)^T X^{(k)} + w_0 = \sum_{i=1}^{m} \alpha_i y^{(i)} X^{(i)} \cdot X^{(k)} + w_0$$

- Potrebna je vrednost skalarnih proizvoda vektora tj. mera njihove sličnosti

Kernel funkcije

- Postoje funkcije koje za zadate vektore $X, Z \in \mathbb{R}^n$ implicitno računaju njihov skalarni proizvod u nekom prostoru sa više dimenzija $\mathbb{R}^q$
- To računanje se ostvaruje *bez* eksplicitnog transformisanja vektora X i Z u prostor $\mathbb{R}^q$
- Takve funkcije se nazivaju *kernel funkcijama*

$$K(X, Z) = \phi(X) \cdot \phi(Z)$$

- Korišćenjem kernel funkcija skup podataka se implicitno prebacuje u višedimenzioni prostor

$$\mathbb{R}^n \longrightarrow \mathbb{R}^q, \qquad q > n$$

- Pri tome se ne troši nikakva dodatna računarska memorija a cena računanja se svodi na računanje $K(X, Z)$ – kernelski trik (engl. *kernel trick*)

Kernel funkcije

- Ovo omogućava da SVM efikasno nauči nelinearne granice razdvajanja zamenom skalarnih proizvoda vektora odabranom kernel funkcijom:

$$h\left(x^{(k)}\right) = \sum_{i=1}^{m} \alpha_i y^{(i)} \phi\left(X^{(i)}\right) \cdot \phi\left(X^{(k)}\right) + w_0$$

$$= \sum_{i=1}^{m} \alpha_i y^{(i)} K\left(X^{(i)}, X^{(k)}\right) + w_0$$

- Kernel funkcija $K\left(X^{(i)}, X^{(k)}\right)$ se sada koristi kao mera sličnosti vektora

Kernel funkcije

- Ono što se eksplicitno bira je kernel funkcija K, a ne funkcija preslikavanja ϕ
  - Funkcija preslikavanja ϕ je implicitno definisana odabirom kernel funkcije K
  - Eksplicitne reprezentacije podataka u višedimenzionalnom prostoru - $\phi(X)$ - se ne konstruišu
- Kernel funkcija mora da ispunjava uslove da bi mogla da se koristi kao mera sličnosti vektora:
  - $K(X, X) = 1$
  - $0 \leq K(X, Z) \leq 1$
  - $K(X, Z) = K(Z, X)$

Neki često korišćeni tipovi kernel funkcija

- Linearni kernel – $K(X, Z) = X \cdot Z$
- Polinomijalni kerneli
- Kerneli sa radijalnom osnovom
- Kerneli za strukturirane podatke
  - String kerneli
  - Kerneli stabla
  - Grafovski kerneli

Polinomijalni kerneli

- Imaju oblik:

$$K(X, Z) = (c + \gamma X \cdot Z)^d$$

- Za $d = 1,\ c = 0,\ \gamma = 1$ dobija se već prikazan model bez kernela tj. linearni kernel
- Često se koristi i kvadratni kernel koji se dobija za $d = 2$
- Polinomijalni kerneli omogućavaju modeliranje kombinacija odlika, sve do stepena polinoma

Polinomijalni kerneli

- Primer – funkcija preslikavanja ϕ polinomijalnog kernela za $n = 2$, $d = 2,\ c = 0,\ \gamma = 1$:

$$K(X,Z) = (X \cdot Z)^2 = \big((x_1, x_2) \cdot (z_1, z_2)\big)^2 = (x_1 z_1 + x_2 z_2)^2$$
$$= x_1^2 z_1^2 + 2 x_1 z_1 x_2 z_2 + x_2^2 z_2^2$$
$$= \left(x_1^2, \sqrt{2} x_1 x_2, x_2^2\right) \cdot \left(z_1^2, \sqrt{2} z_1 z_2, z_2^2\right)$$
$$= \phi(X) \cdot \phi(Z)$$

- Sledi da je za navedene vrednosti parametara funkcija preslikavanja polinomijalnog kernela:

$$\phi(X) = \left(x_1^2, \sqrt{2} x_1 x_2, x_2^2\right)$$

Kerneli sa radijalnom osnovom

- Kerneli sa radijalnom osnovom (engl. *radial basis function kernels* – *RBF kernels*) imaju oblik:

$$K(X, Z) = \phi(\|X - Z\|_2)$$

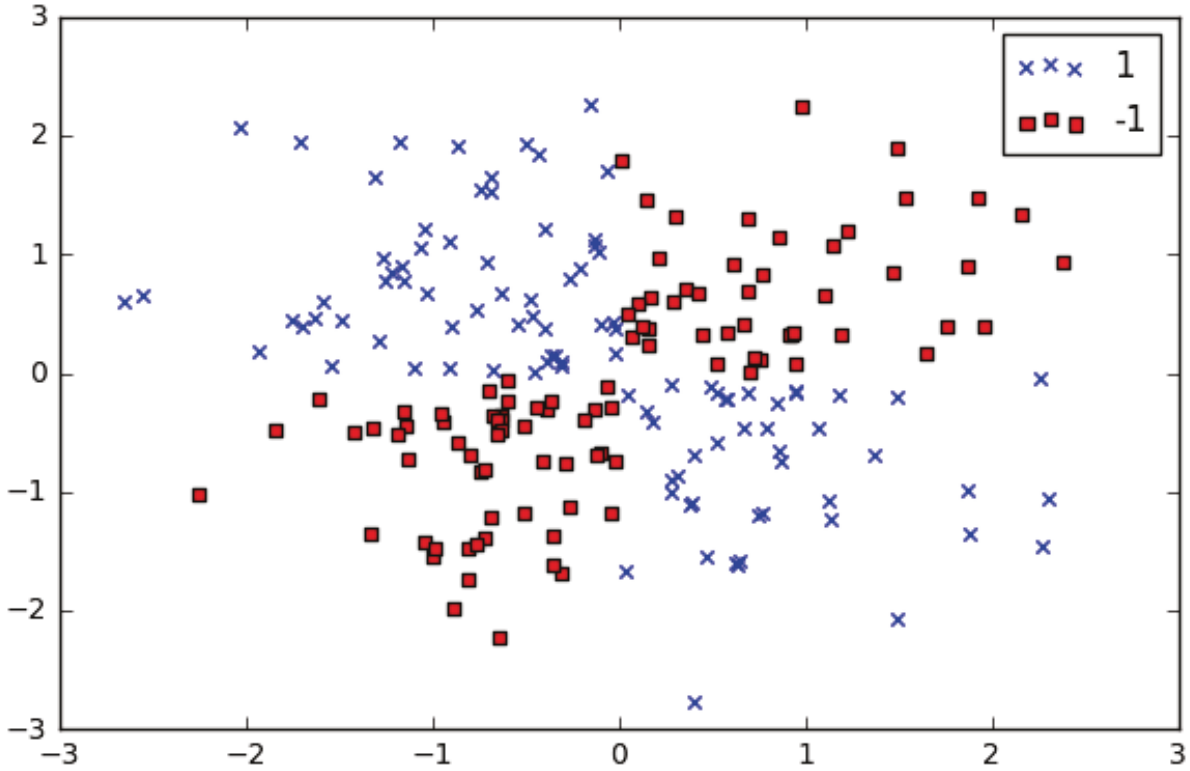
- Najčešće korišćena varijanta kernel funkcije sa radijalnom osnovom je Gausova, u obliku:

$$K(X, Z) = e^{-\frac{\|X-Z\|_2^2}{2\sigma^2}} = e^{-\gamma\|X-Z\|_2^2}$$

- gde je $\gamma = \frac{1}{2\sigma^2}$ preciznost, a σ^2 širina Gausove krive (engl. *bandwidth*)
- Kernel funkcije sa radijalnom osnovom su ekvivalentne mapiranju u prostor sa beskonačno mnogo dimenzija

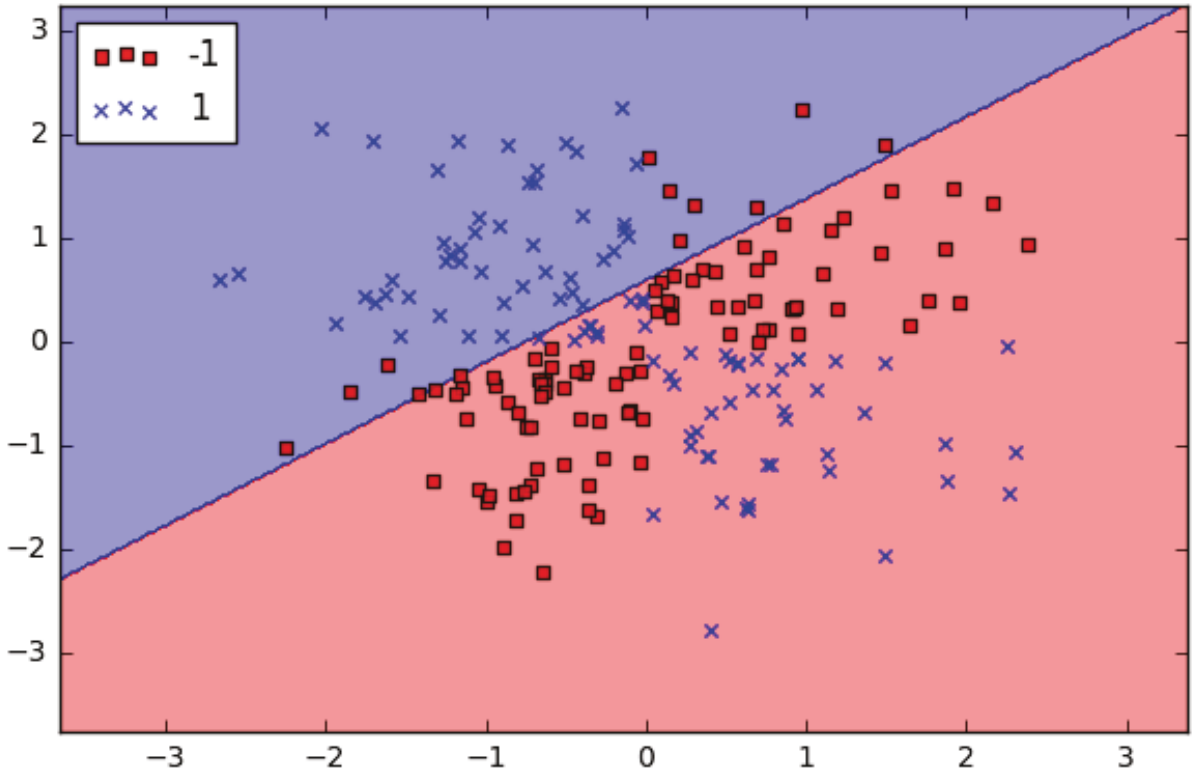
Gausov RBF kernel

- Parametar γ kod Gausovog RBF kernela ima veliki uticaj na ponašanje modela
- γ je inverzno širini Gausove krive
  - Jako mala vrednost γ znači da je Gausova kriva izuzetno široka i da svaki potporni vektor X (donekle) ima efekat na klasifikaciju svih podataka
    - Izlaz hipoteze se menja vrlo polako sa promenom vrednosti ulaza – veliko sistematsko odstupanje (engl. *bias*) modela tj. nedovoljna prilagođenost modela podacima
  - Jako velika vrednost γ znači da je Gausova kriva izuzetno uska i da svaki potporni vektor X ima efekat na klasifikaciju samo onih podataka koji se nalaze u njegovoj neposrednoj blizini
    - Izlaz hipoteze se menja izuzetno brzo sa promenom vrednosti ulaza – velika varijansa modela tj. preterana prilagođenost modela podacima



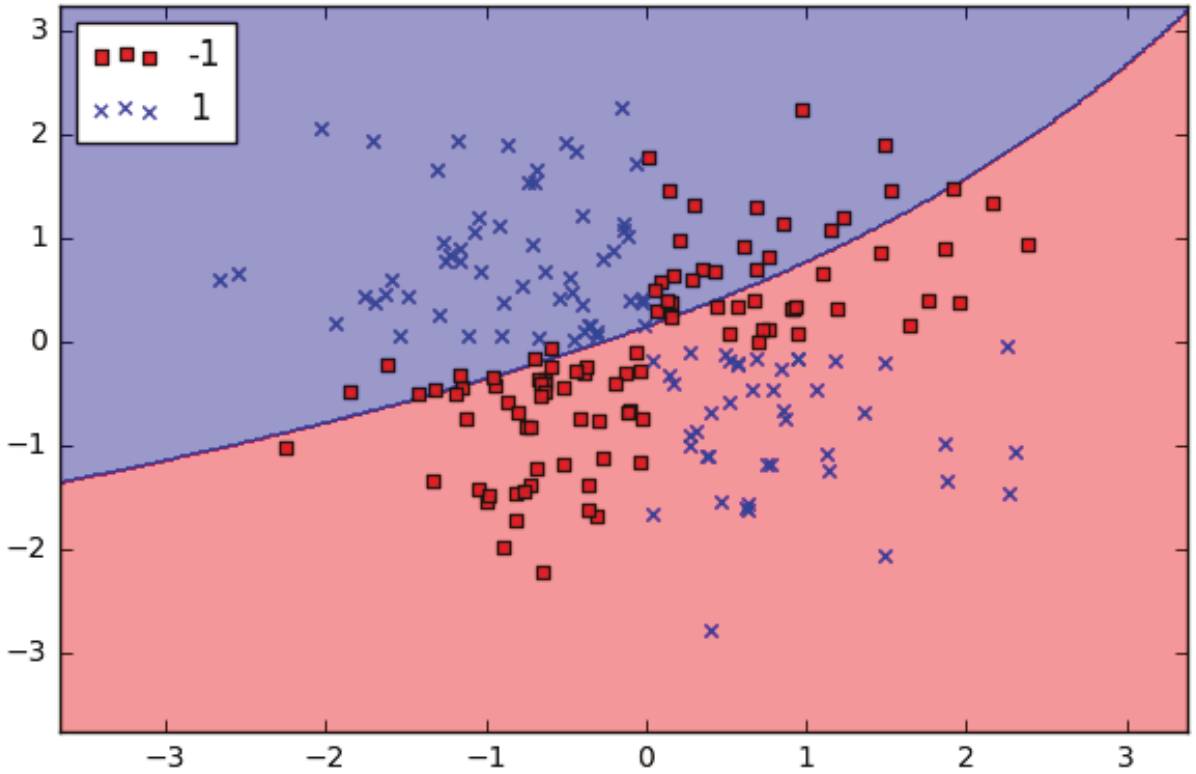
Ilustracija ponašanja različitih SVM kernela – raspodela podataka

Slika preuzeta sa: http://chrisalbon.com/machine_learning/support_vector_machines/svc_parameters_using_rbf_kernel/



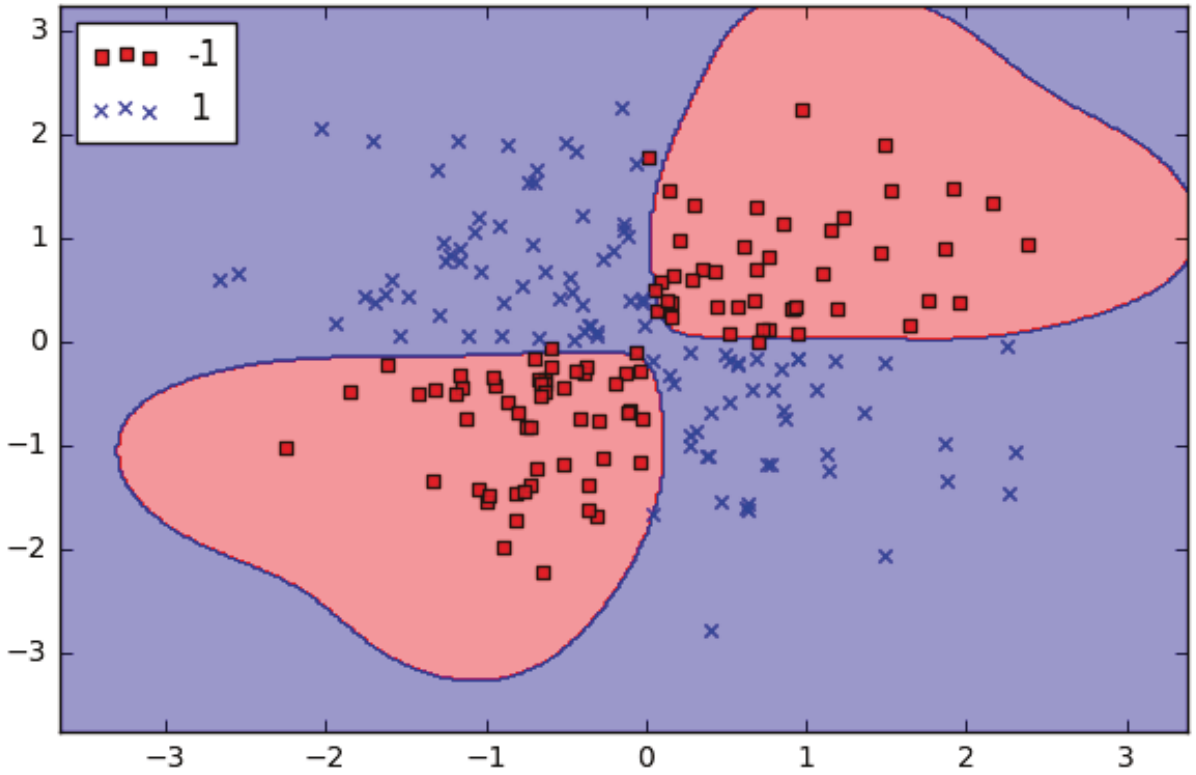
Ilustracija ponašanja različitih SVM kernela – linearni kernel (tj. SVM bez kernela), $C = 1$

Slika preuzeta sa: http://chrisalbon.com/machine_learning/support_vector_machines/svc_parameters_using_rbf_kernel/



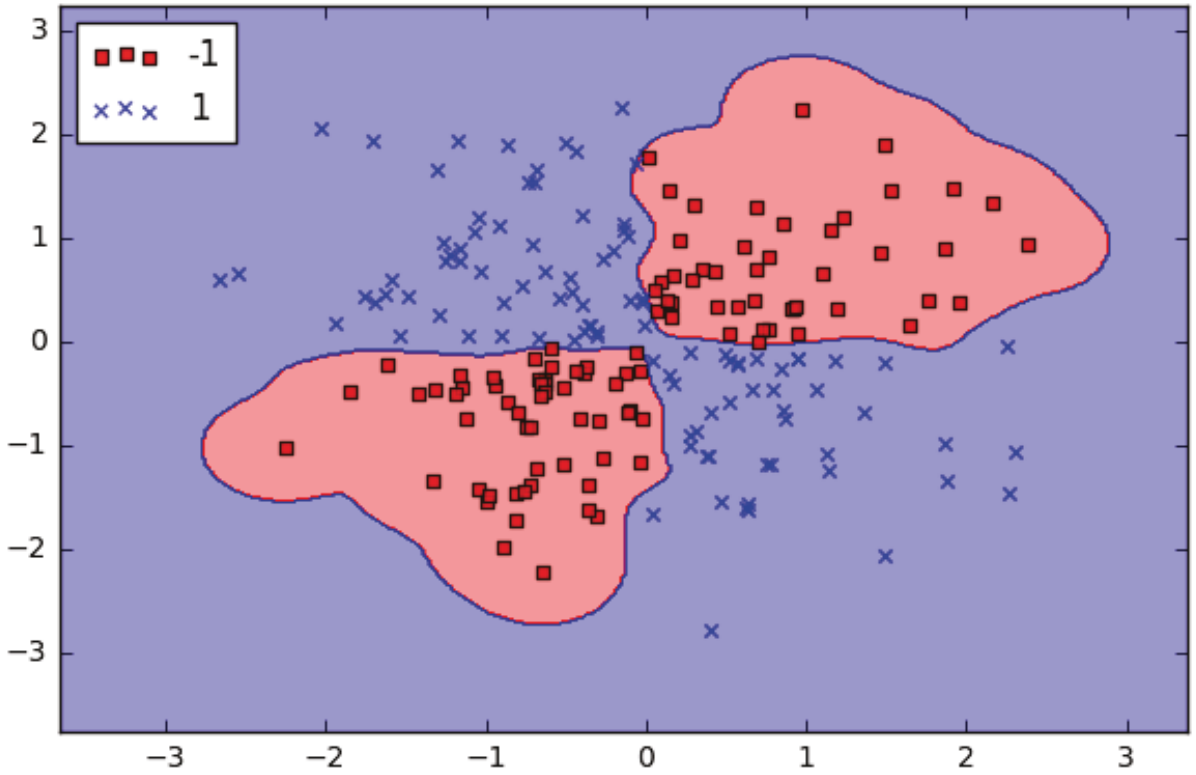
Ilustracija ponašanja različitih SVM kernela – Gausov RBF kernel, $\gamma = 0.01$, $C = 1$

Slika preuzeta sa: http://chrisalbon.com/machine_learning/support_vector_machines/svc_parameters_using_rbf_kernel/



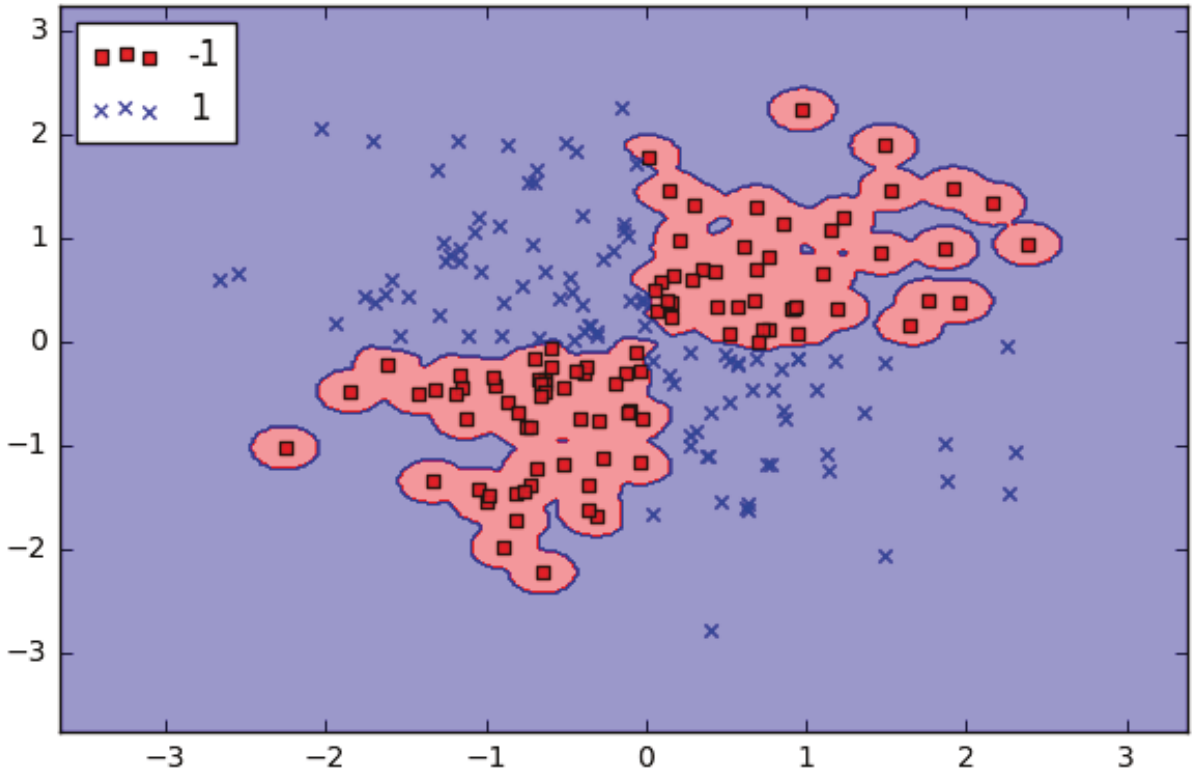
Ilustracija ponašanja različitih SVM kernela – Gausov RBF kernel, $\gamma = 1$, $C = 1$

Slika preuzeta sa: http://chrisalbon.com/machine_learning/support_vector_machines/svc_parameters_using_rbf_kernel/



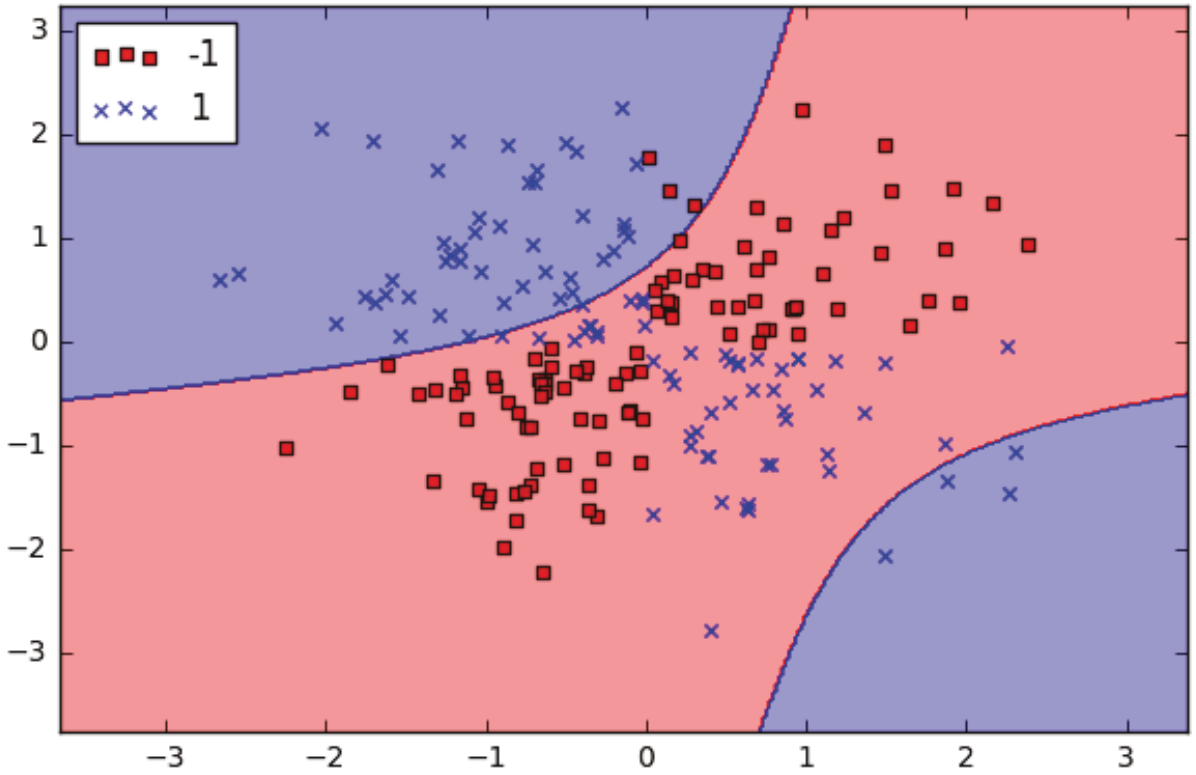
Ilustracija ponašanja različitih SVM kernela – Gausov RBF kernel, $\gamma = 10$, $C = 1$

Slika preuzeta sa: http://chrisalbon.com/machine_learning/support_vector_machines/svc_parameters_using_rbf_kernel/



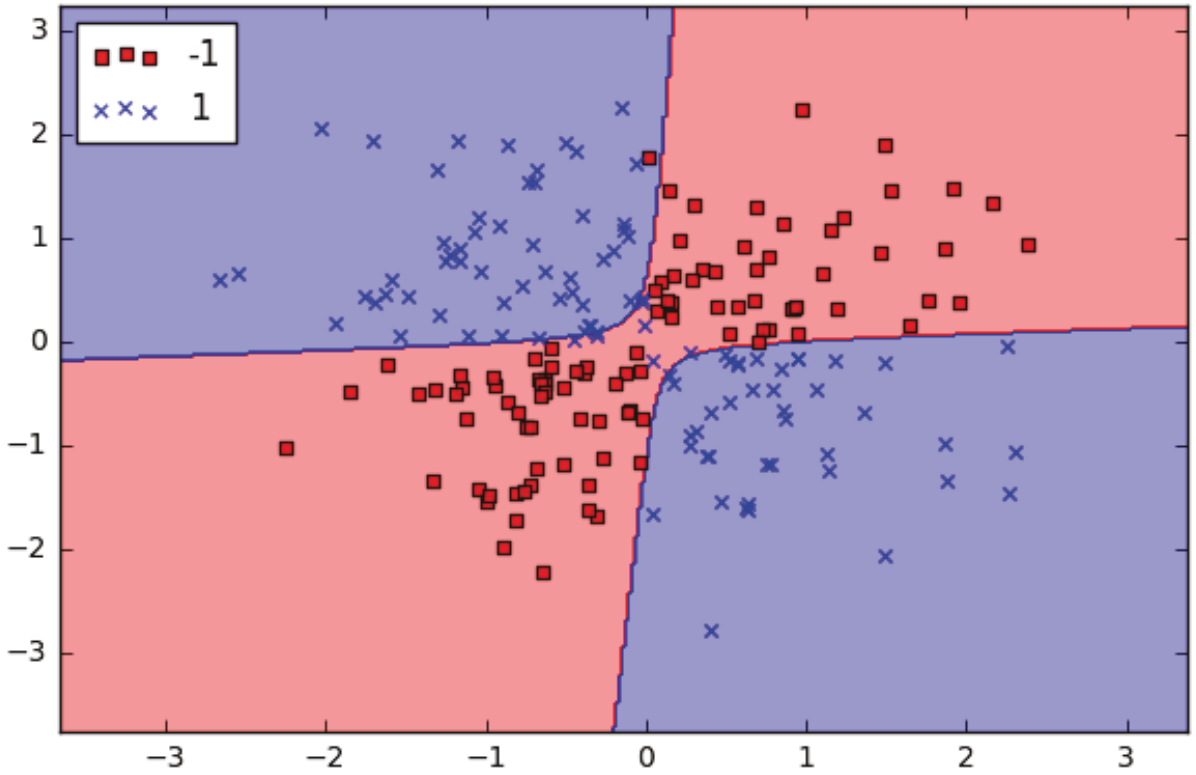
Ilustracija ponašanja različitih SVM kernela – Gausov RBF kernel, $\gamma = 100$, $C = 1$

Slika preuzeta sa: http://chrisalbon.com/machine_learning/support_vector_machines/svc_parameters_using_rbf_kernel/



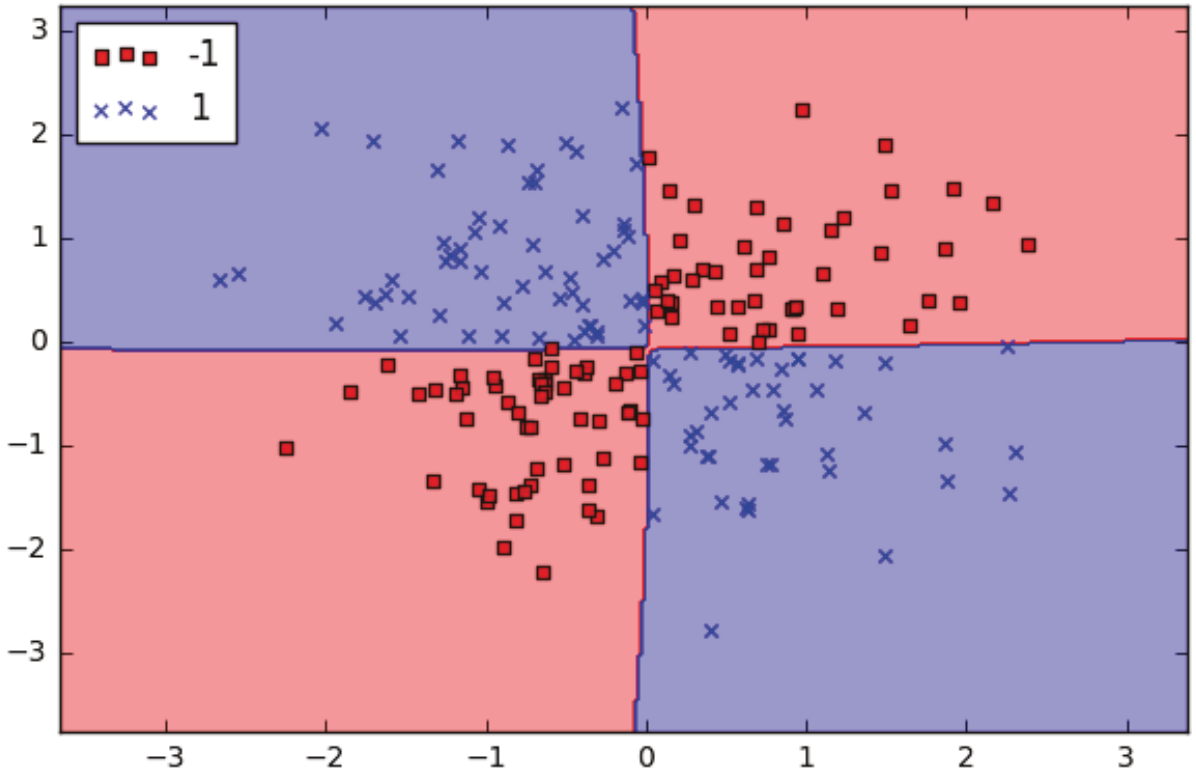
Ilustracija ponašanja različitih SVM kernela – Gausov RBF kernel, $\gamma = 0.01$, $C = 10$

Slika preuzeta sa: http://chrisalbon.com/machine_learning/support_vector_machines/svc_parameters_using_rbf_kernel/



Ilustracija ponašanja različitih SVM kernela – Gausov RBF kernel, $\gamma = 0.01$, $C = 1000$

Slika preuzeta sa: http://chrisalbon.com/machine_learning/support_vector_machines/svc_parameters_using_rbf_kernel/



Ilustracija ponašanja različitih SVM kernela – Gausov RBF kernel, $\gamma = 0.01$, $C = 100000$

Slika preuzeta sa: http://chrisalbon.com/machine_learning/support_vector_machines/svc_parameters_using_rbf_kernel/

Video prikazi delovanja različitih kernela

- Polinomijalni kernel
  - http://www.youtube.com/watch?v=3liCbRZPrZA
  - http://www.youtube.com/watch?v=ndNE8he7Nnk
- Gausov RBF kernel
  - http://www.youtube.com/watch?v=9NrALgHFwTo
  - http://www.youtube.com/watch?v=RtajnP3tGsc

Odabir kernela

- Odabiranje najpogodnijeg kernela zavisi od konkretnog zadatka i može da bude netrivijalno
- Bez obzira koji kernel je odabran, njegovi parametri su hiperparametri modela
  - I izbor kernela može da se tretira kao hiperparametar
- Vrednosti hiperparametara je neophodno optimizovati da bi se dobile dobre performanse klasifikacije
  - To se čini putem (ugnežđene) unakrsne validacije
  - Vrednosti različitih hiperparametara mogu da interaguju, tako da je potrebno pretraživanje po mreži (engl. *grid search*)

SVM optimizacija u primarnom domenu

- Funkciju greške je moguće minimizovati i u primarnom domenu, bez korišćenja Lagranžovih multiplikatora i prelaska u dualni domen
- Dualni pristup može biti spor kada je količina podataka u skupu za obučavanje jako velika, pošto je tada obično i broj potpornih vektora veliki
- Ograničenje u primarnom domenu – nepostojanje kernela
  - Moguće je rešavanje samo za linearni SVM
- Primenjuju se naprednije tehnike numeričke optimizacije, kao što je subgradijentni spust
  - Često se traži aproksimacija optimalnog rešenja, a ne tačno rešenje
- Kada je broj odlika znatno veći od broja podataka u skupu za obučavanje može biti preporučljivije da se optimizacija radi u dualnom domenu

Višeklasna klasifikacija

- Metoda potpornih vektora se može primeniti i u slučaju prisustva većeg broja klasa, i to kombinovanjem rezultata većeg broja nezavisnih binarnih klasifikacija
  - Princip „jedan nasuprot svima“ (engl. *one-versus-all*) / „jedan nasuprot ostalima“ (engl. *one-versus-rest*)
    - Podatak se svrstava u onu klasu čiji binarni klasifikator klasifikuje dati podatak u tu klasu sa najvećom udaljenošću između njegove hiperravni razdvajanja i podatka (umesto najveće verovatnoće pripadanja određenoj klasi u slučaju logističke regresije)
  - Princip „jedan nasuprot jednom“ (engl. *one-versus-one*)
- Postoje i pristupi koji istovremeno modeluju prisustvo više klasa
  - Crammer & Singer algoritam

Prednosti i mane metode potpornih vektora

- Prednosti
  - Veoma dobre performanse na širokom spektru problema
  - Metoda sama odabira koji podaci su potporni vektori i koliko ih ima
  - Manja sklonost *overfitting*-u od mnogih drugih metoda
  - Interakcije odlika se mogu implicitno modelovati preko kernela
- Mane
  - SVM može da bude primetno sporiji od drugih metoda, u zavisnosti od broja odlika i količine podataka u skupu za obučavanje
    - Korišćenje kernela dodatno usporava rad algoritma
  - Izlaz nije probabilističkog tipa
    - Moguća probabilistička procena preko udaljenosti od hiperravni razdvajanja
  - Varijanta sa čvrstom marginom je osetljiva na *outlier*-e