

Metod potpornih vektora (eng. support vector machine)

Metod potpornih vektora (eng. support vector machine), nadalje samo SVM, ima raznoliku primenu u klasifikacionim problemima mašinskog učenja i dosta je matematički potkrepljen. Može se koristiti i za regresiju, ali se mnogo češće koristi za klasifikaciju.

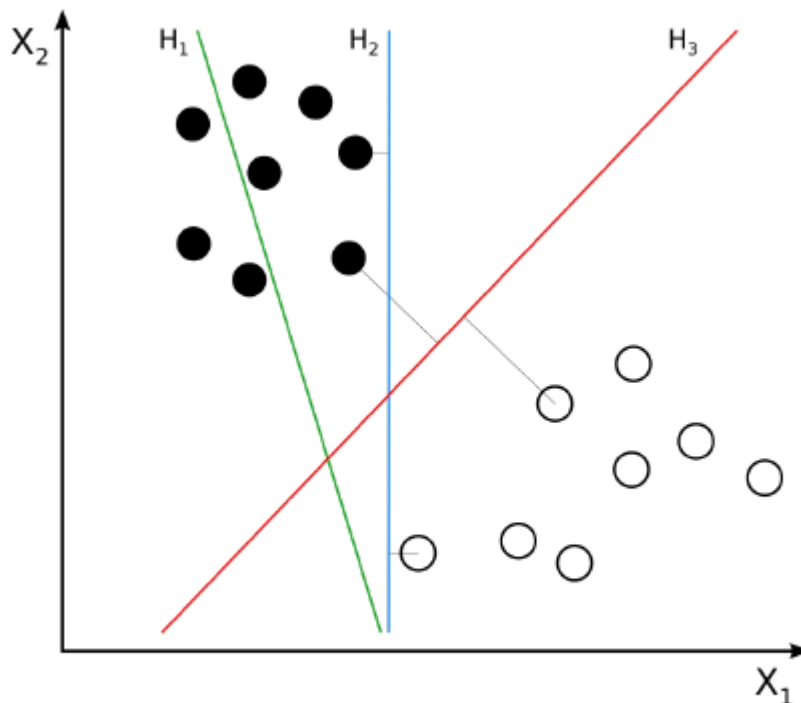
SVM za klasifikaciju

Preporučujem vam da pogledate sledeći snimak na kojem je ovaj metod ilustrativno objašnjen:

<https://www.youtube.com/watch?v=efR1C6CvhmE>

Pretpostavimo da je pred nama zadatak binarne klasifikacije i da u skupu za obučavanje imamo n klasifikovanih tačaka x_1, \dots, x_n iz prostora R^d . Radi jednostavnosti, ilustrovaćemo slučaj za $d = 2$. Pretpostavićemo dodatno i da je tačke moguće razdvojiti pravom tako da se sa različitih strana prave nalaze tačke koje pripadaju različitim klasama. Intuicija nam kaže da bi novu tačku trebalo klasifikovati kao i one koje se nalaze sa iste strane prave kao i posmatrana tačka. Dakle, jedino što nam preostaje da uradimo je da odredimo granicu, odnosno pomenutu pravu kojom ćemo podeliti prostor. To možemo učiniti na više načina, ali nam se ipak čini da su neke prave bolja opcija od drugih. Razlog tome je taj što smo za tačke koje su na što manjoj udaljenosti od granice sve manje sigurni da su ispravno klasifikovane. Zato je cilj da rastojanje prave od tačaka koje su među predstavnicima svoje klase njoj najbliže bude što veće, pa za optimalnu

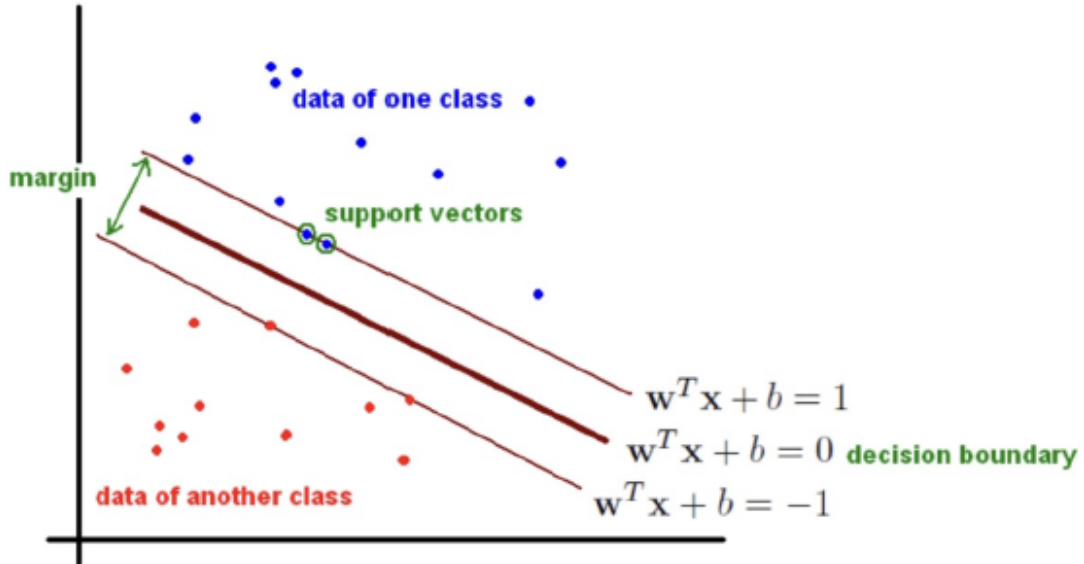
pravu biramo onu koja je na najvećoj udaljenosti od njoj najbliže tačke podataka.



prava H_1 ne razdvaja tačke; prava H_2 razdvaja, ali ne na dovoljno uverljiv način; prava H_3 je optimalna.

Crna i bela tačka koje su najbliže pravoj H_3 zovu se potporni vektori i po njima je ovaj metod dobio ime. Kako ne želimo da se ograničimo na dvodimenzionalni slučaj, potrebno je da uvodnu priču malo formalizujemo. Naime, prava je u ravni hiperravan, odnosno potprostor dimenzije za 1 manje od dimenzije posmatranog prostora. Slično, hiperravan na pravoj je tačka, a hiperravan u prostoru je ravan. Naš cilj je, dakle, pronalaženje optimalne hiperravni u prostoru proizvoljne dimenzije. Kako je lakše interpretirati dešavanja u prostoru male dimenzije, slikom ostajemo u ravni, ali da bi se naredne formule prenele i na veću dimenziju, umesto poznate jednačine prave $ax + by + c = 0$ koristićemo zapis sa skalarnim proizvodom: $\omega^T x + b = 0$ gde su ω i x vektori kolone odgovarajuće dimenzije. (Podsećanje: prava je određena jednom tačkom M i vektorom normale ω . Za tačke koje leže na njoj važi: $\omega^T(x - M) = 0$, tj. $\omega^T x - \omega^T M = 0$). Optimizacioni problem se svodi na nalaženje koeficijenata b i ω takvih da je rastojanje između hiperravni određenih potpornim vektorima svojih klasa najveće. Za detalje pratiti

sledeću sliku.



Optimizacija

Pretpostavimo da smo odredili graničnu pravu. Prave paralelne pravoj zadatoj sa $\omega^T x + b = 0$ koje su na jednakom rastojanju od nje sa suprotnih strana imaju jednačine: $\omega^T x + b = c$ i $\omega^T x + b = -c$. Zbog računskih olakšica koje slede, smatraćemo da je $c = 1$ i raditi sa jednačinama sa slike. Važno je znati sledeće:

- za tačke van pojasa označenog kao margina (kao i za one na njemu) važi $\omega^T x + b \geq 1$ (plave tačke) ili $\omega^T x + b \leq -1$ (crvene tačke);
- granična prava deli prostor tako da za tačke s jedne strane važi $\omega^T x + b > 0$, a za one sa druge: $\omega^T x + b < 0$. Klase tačaka x_i označićemo sa $y_i = 1$ u prvom slučaju, odnosno sa $y_i = -1$ u drugom slučaju.

Pod ovim oznakama za sve tačke sa slike važi: $y_i(\omega^T x_i + b) \geq 1$

Izračunajmo sada širinu margine koju želimo da maksimizujemo. Uočimo proizvoljnu tačku p za koju važi $\omega^T p + b = -1$. Njoj najbliža tačka q takva da je $\omega^T q + b = 1$ ima oblik $q = p + \lambda\omega$, pa je rastojanje između ovih tačaka jednako $|\lambda||\omega|$. Uz malo računa dobijamo: $\omega^T q + b = \omega^T p + \lambda|\omega|^2 + b$ odnosno $1 = \lambda|\omega|^2 - 1$, odakle je $\lambda = \frac{2}{|\omega|^2}$.

Sledi: $\lambda|\omega| = \frac{2}{|\omega|}$. Dakle, optimizacioni problem je: minimizovati

$$\frac{||w||}{2}$$

pri uslovu $y_i(\omega^T x_i + b) \geq 1, \forall i \in \{1, 2, \dots, n\}$.

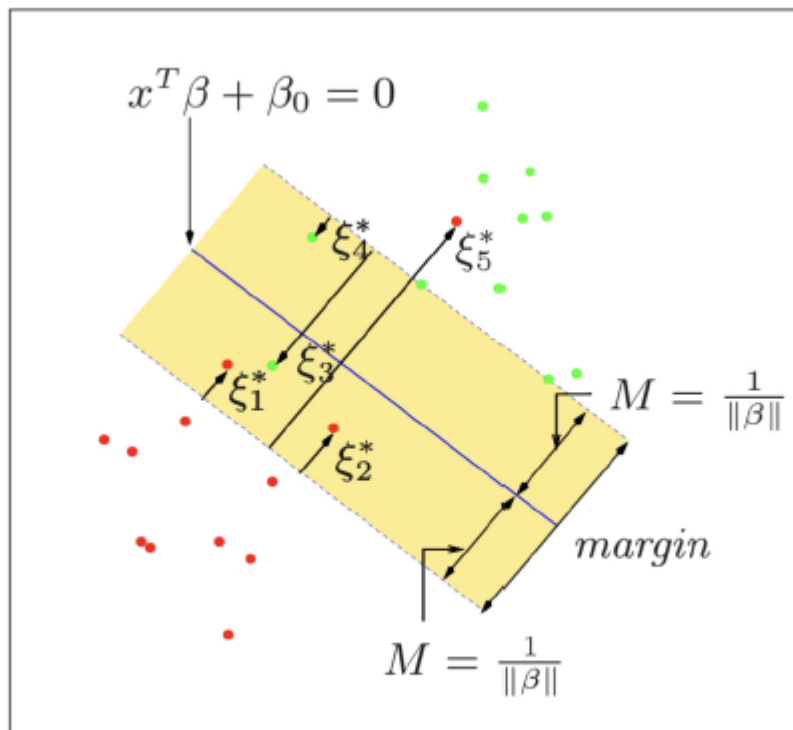
Dodatni uslov obezbeđuje da se sve tačke nalaze van pojasa određenog hiperravnima potpornih vektora. Razlog zbog kog računamo minimum umesto maksimuma je taj što su mnogi optimizacioni algoritmi konstruisani upravo za probleme minimizacije.

Postoje li problemi s ovim pristupom?

Pored očigledne činjenice da su tačke retko kad linearno razdvojive, ovakvim načinom izbora granične hiperravni smo se previše prilagodili podacima koje imamo. Zato je ideja da dozvolimo nekim tačkama da se nađu sa pogrešne strane hiperravni određene potpornim vektorima svoje klase, ali ne previše. To činimo uvođenjem novih promenljivih $\xi_i \geq 0, i \in \{1, 2, \dots, n\}$ i dodavanjem uslova: $y_i(\omega^T x_i + b) \geq 1 - \xi_i, \forall i \in \{1, 2, \dots, n\}$. Ako je $\xi_i = 0$, tačka x_i je sa prave strane. Za $\xi_i > 1$ ona može biti čak i sa druge strane granične prave. Novi optimizacioni problem sada izgleda ovako: minimizovati

$$\frac{\|w\|}{2} + C \sum_{i=1}^n \xi_i$$

pri uslovu: $y_i(\omega^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0, \forall i \in \{1, 2, \dots, n\}$.

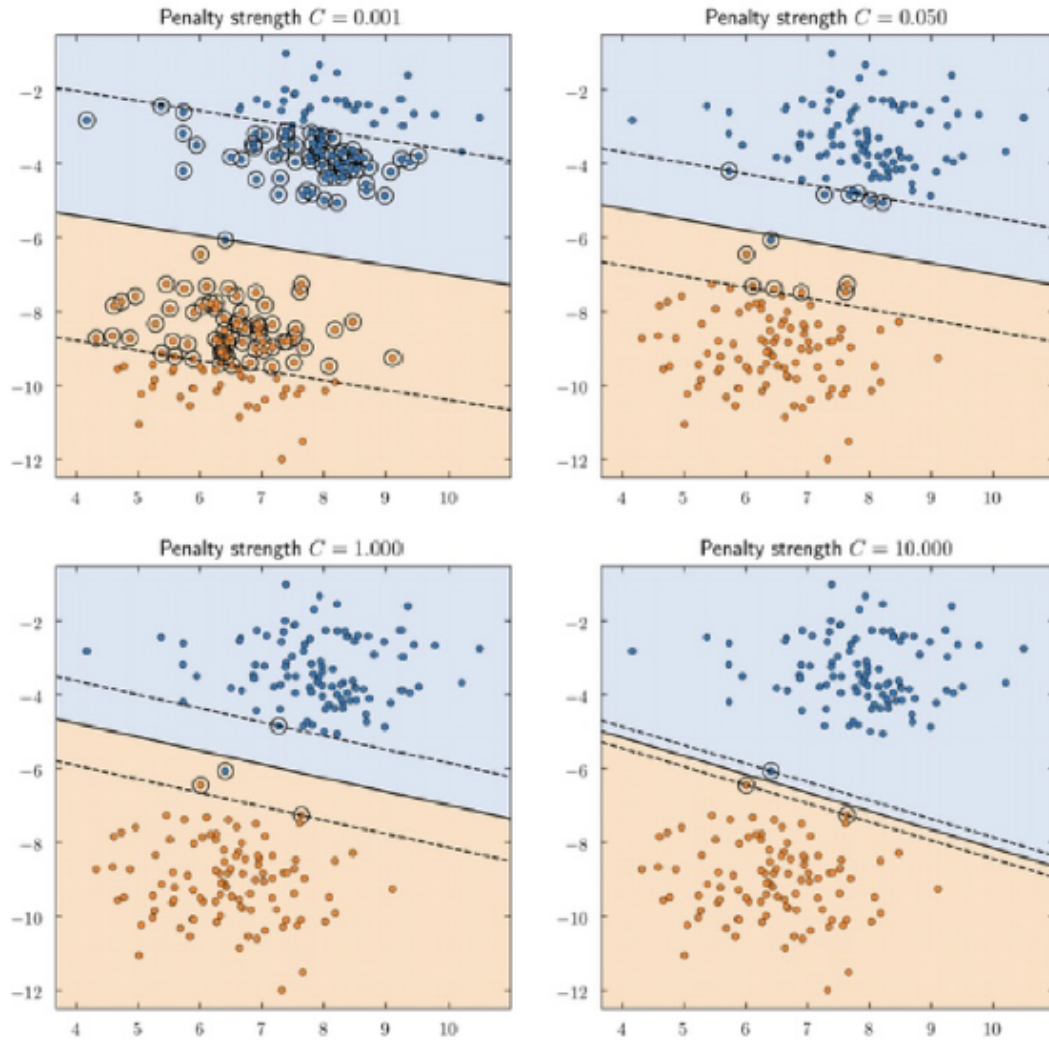


za tačke ξ_i^* vazi $\xi_i > 0$ za $i \in \{1, 2, 4\}$, odnosno $\xi_i > 1$ za $i \in \{3, 5\}$

Hiperparametar $C > 0$ određuje koliko se značajja pridaje greškama, odnosno, koliko suma $\sum_{i=1}^n \xi_i$ može da varira.

Za male vrednosti C član $C \sum_{i=1}^n \xi_i$ u izrazu koji minimizujemo postaje zanemarljiv, pa greške koje pravimo mogu biti jako velike. Što je vrednost C veća, to je član $C \sum_{i=1}^n \xi_i$ dominantniji, a s obzirom da tražimo minimum funkcije, suma $\sum_{i=1}^n \xi_i$ mora biti što manja, pa samim tim su ξ_i bliži nuli. Dakle, za velike vrednosti C greške su minimalne i granična prava se sve više približava onoj iz prvobitne ideje. Ovakav metod se zove metod potpornih vektora sa mekim pojasom (eng. soft margin). Na sledećoj slici se

moгу videti izbori granične hiperravni za različite vrednosti C .



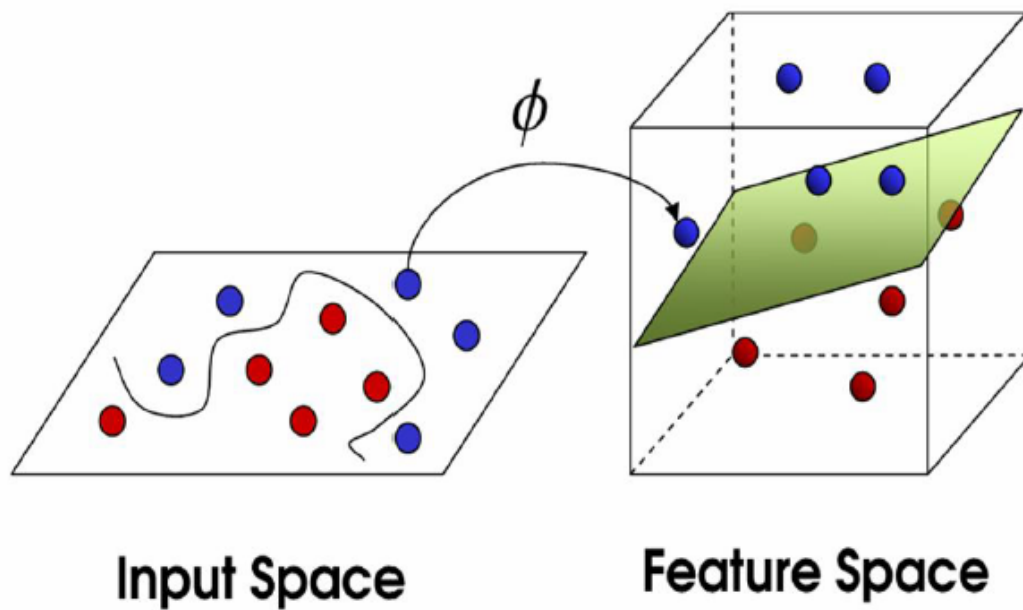
Šta raditi ako podaci nisu linearno razdvojnivi?

Trik koji ćemo sada primeniti može se iskoristiti u bilo kom klasifikatoru.

Uvešćemo preslikavanje $\phi : R^d \rightarrow R^m$ takvo da su slike tačaka iz prostora R^d linearno

razdvojive u R^m njegovom hiperravnini.

Principle of Support Vector Machines (SVM)



Često korišćena jezgra

Polinomijalno jezgro

$$k(x_i, x_j) = (x_i^T x_j + 1)^d$$

Gausovo jezgro

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

Gausovo radijalno jezgro (RBF)

$$k(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|^2), \gamma > 0$$

Laplasovo radijalno jezgro

$$k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|}{\sigma}\right)$$

Sigmoidno jezgro

$$k(x_i, x_j) = \tanh(\alpha x_i^T x_j + c)$$

Parametri jezgra su hiperparametri modela i određuju se na validacionom skupu.

Primena SVM

Prepoznavanje lica

Kategorizacija teksta - npr. detekcija spama

Klasifikacija ručno pisanog teksta

Prepoznavanje govora

Predviđanje cena akcija

Sledeći kod je većinski preuzet od asistenata Anđelke Zečević i Milana Čugurovića sa časova vežbi iz Mašinskog učenja.

```
In [16]: import numpy as np
import pandas as pd
```

```
In [17]: from matplotlib import pyplot as plt
```

```
In [18]: from sklearn import svm
from sklearn import model_selection
from sklearn import metrics
from sklearn import datasets
from sklearn import preprocessing
```

U radu ćemo koristiti Viskonsis skup podataka za klasifikaciju tumora na benigne i maligne koji smo koristili i u primeru sa logističkom regresijom.

```
In [19]: data = datasets.load_breast_cancer()
```

Podsetimo se da skup podataka ima 30 atributa numeričkog tipa i ciljnu promenljivu sa vrednostima 0 i 1.

```
In [20]: len(data.feature_names)
```

```
Out[20]: 30
```

```
In [21]: data.feature_names
```

```
Out[21]: array(['mean radius', 'mean texture', 'mean perimeter', 'mean area',
               'mean smoothness', 'mean compactness', 'mean concavity',
               'mean concave points', 'mean symmetry', 'mean fractal dimension',
               'radius error', 'texture error', 'perimeter error', 'area error',
               'smoothness error', 'compactness error', 'concavity error',
               'concave points error', 'symmetry error',
               'fractal dimension error', 'worst radius', 'worst texture',
               'worst perimeter', 'worst area', 'worst smoothness',
               'worst compactness', 'worst concavity', 'worst concave points',
               'worst symmetry', 'worst fractal dimension'], dtype='<U23')
```

```
In [22]: X = data.data
```

```
In [23]: X.shape
```

Out[23]: (569, 30)

```
In [24]: y = data.target
```

Trening i test skup delimo u odnosu 70:30.

```
In [25]: X_train, X_test, y_train, y_test = model_selection.train_test_split(X, y,
test_size = 0.3, random_state = 42, stratify = y)
```

Standardizujemo podatke.

```
In [26]: scaler = preprocessing.StandardScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

Hiperparametri SVM modela su C i γ . C kontroliše jačinu regularizacije, a γ širinu Gausovog radijalnog jezgra (RBF kernela).

```
In [27]: # jačina regularizacije je inverzno proporcionalna parametru C
# C mora biti strogo pozitivan broj
# podrazumevano se koristi kvadrat l2 regularizacije
model = svm.SVC(kernel='rbf', gamma=0.2, C = 2)
```

```
In [28]: model.fit(X_train, y_train)
```

```
Out[28]: SVC
SVC(C=2, gamma=0.2)
```

I na kraju testiramo model koristeći test skup:

```
In [29]: f1_score = metrics.f1_score(y_test, model.predict(X_test))
```

```
In [30]: print('f1-score na test skupu je: ', f1_score)
```

f1-score na test skupu je: 0.9519230769230769