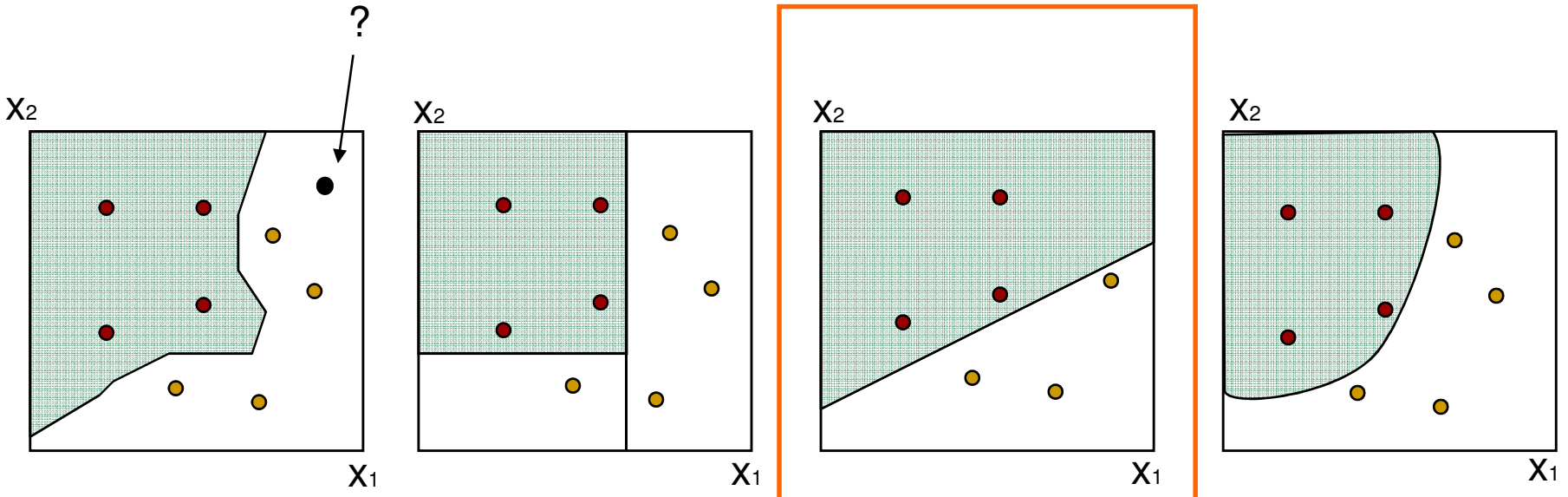
Support Vector Machines

Support Vector Machines

- Predstavujú veľmi dôležitý objav z oblasti strojového učenia
- Rozhodovanie na základe hyperplochy rovnako ako pri perceptróne (priamka v 2D, rovina v 3D, atď.)
- Nájdenie hyperplochy ktorá zabezpečí najväčšie oddelenie dvoch tried vstupných vzorov

Rozhodovacia funkcia

- Realizuje klasifikovanie nového vstupného vzoru **x**:



Nearest Neighbor

Rozhodovacie stromy

Lineárne funkcie

$$g(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$$

Nelineárne funkcie

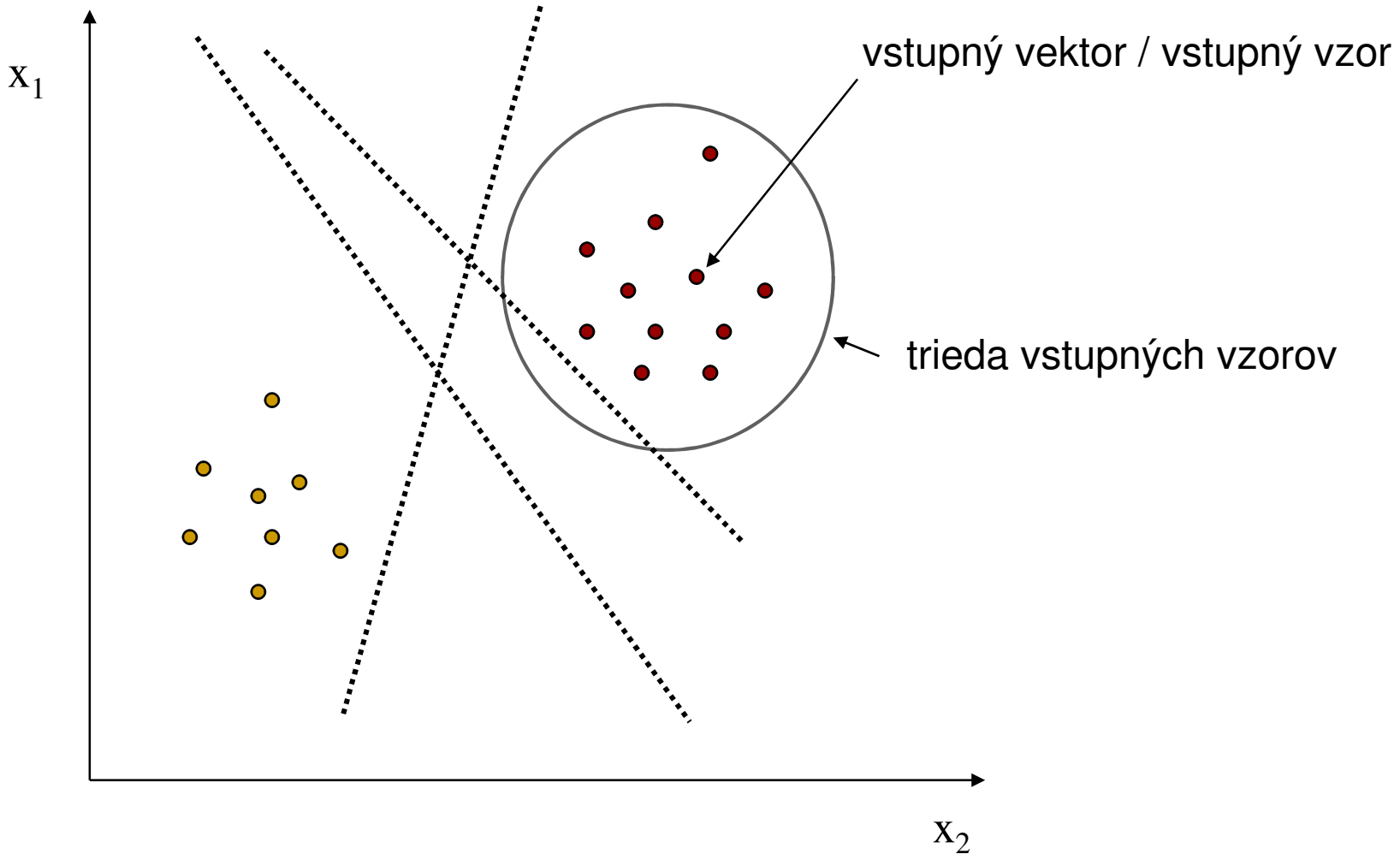
Support Vector Machines

- Definujú **optimálnu** deliacu rovinu - **maximalizovanie hranice medzi triedami**
- Dokážu fungovať aj na lineárne neseparovateľných problémoch – **pomocou penalizácie za zlú klasifikáciu**
- Transformujú vstupy do “priestoru zaujímavých vlastností” (feature space) – **problém je preformulovaný tak aby v sebe transformáciu zahŕňal implicitne**

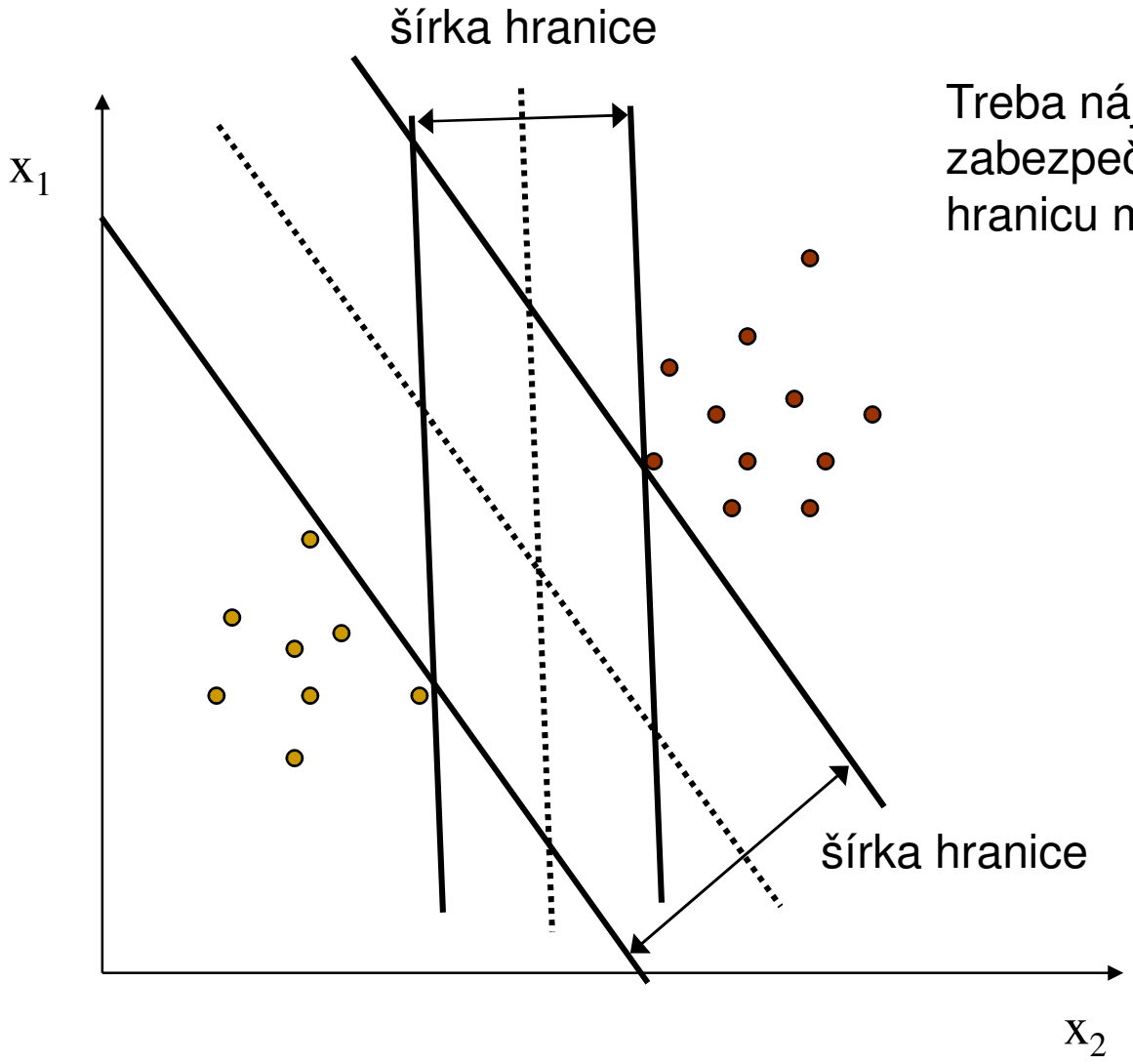
Support Vector Machines

- Definujú **optimálnu** deliacu rovinu - **maximalizovanie hranice medzi triedami**
- Dokážu fungovať aj na lineárne neseparovateľných problémoch – **pomocou penalizácie za zlú klasifikáciu**
- Transformujú vstupy do “priestoru zaujímavých vlastností” (feature space) – **problém je preformulovaný tak aby v sebe transformáciu zahŕňal implicitne**

Ktorá rovina separuje triedy optimálne?

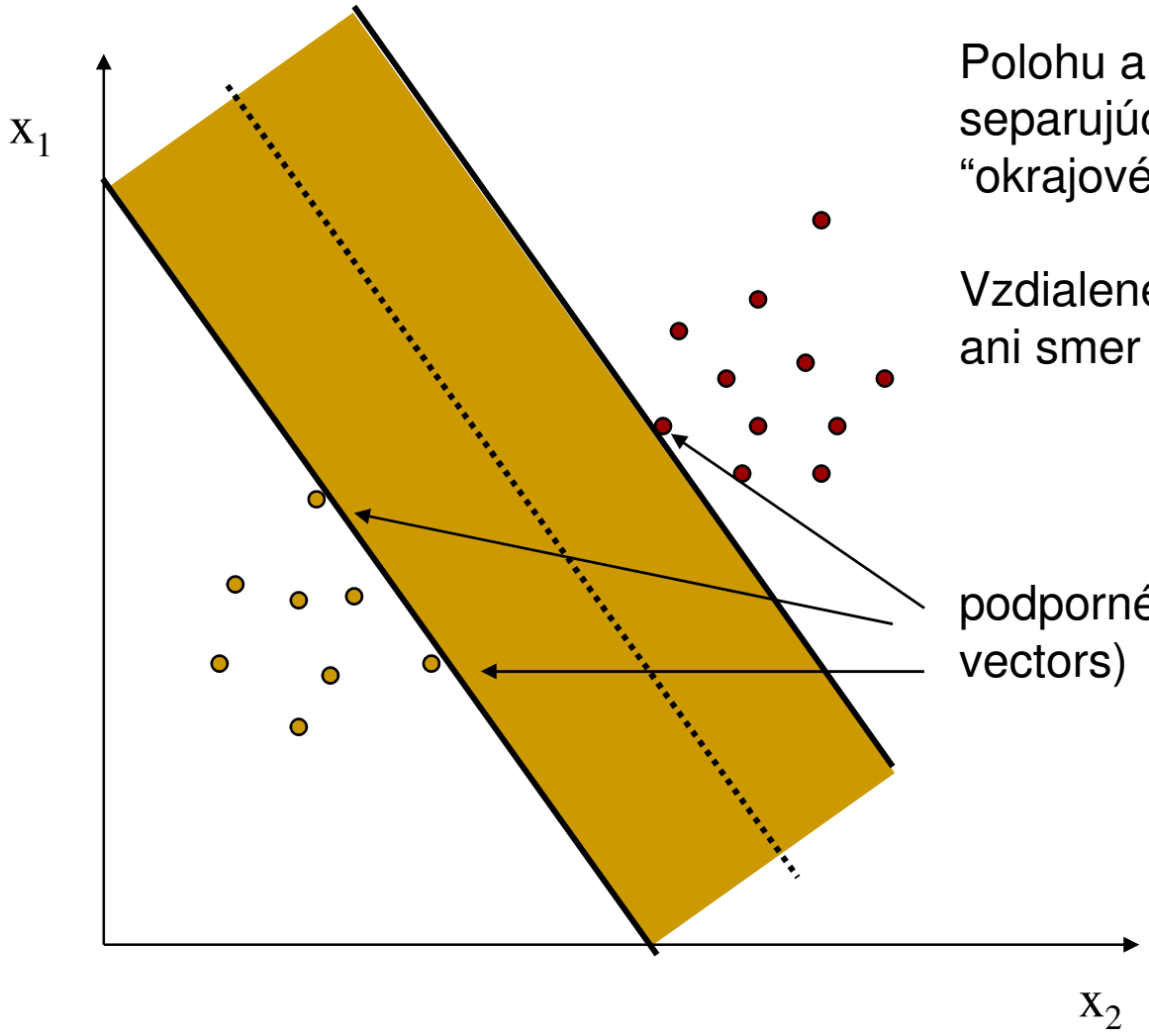


Maximalizovanie hranice medzi triedami



Treba nájsť takú rovinu ktorá zabezpečí najširšiu oddeľujúcu hranicu medzi oboma triedami

Podporné vektory

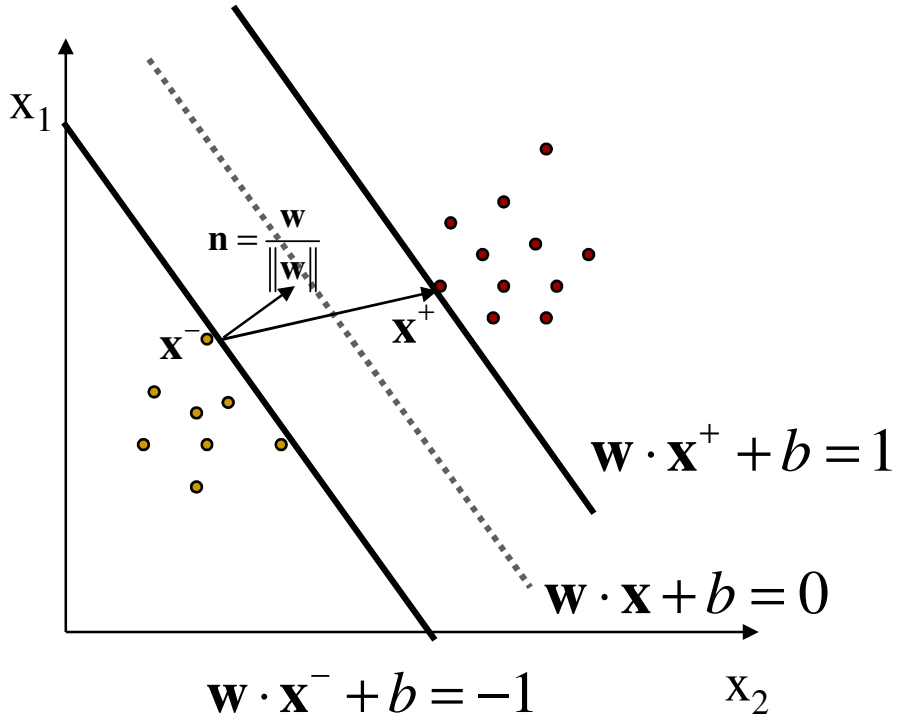


Polohu a smer optimálne separujúcej roviny určujú len "okrajové body" v oboch triedach.

Vzdialenejšie vstupy na jej polohu ani smer nevplývajú

podporné vektory (angl. support vectors)

Formálna definícia problému



Skalárny súčin je definovaný ako *súčin* veľkostí dvoch *vektorov* a kosínusu uhla, ktorý navzájom zvierajú.

Vstupný vzor – vektor x klasifikujeme na základe vektora váh **w** a prahu b.

$$\operatorname{sgn}\left(\mathbf{w}\cdot\mathbf{x}+b\right)$$

Pre body na okraji hranice platí

$$\mathbf{w}\cdot\mathbf{x}^{+}+b=1$$

$$\mathbf{w}\cdot\mathbf{x}^{-}+b=-1$$

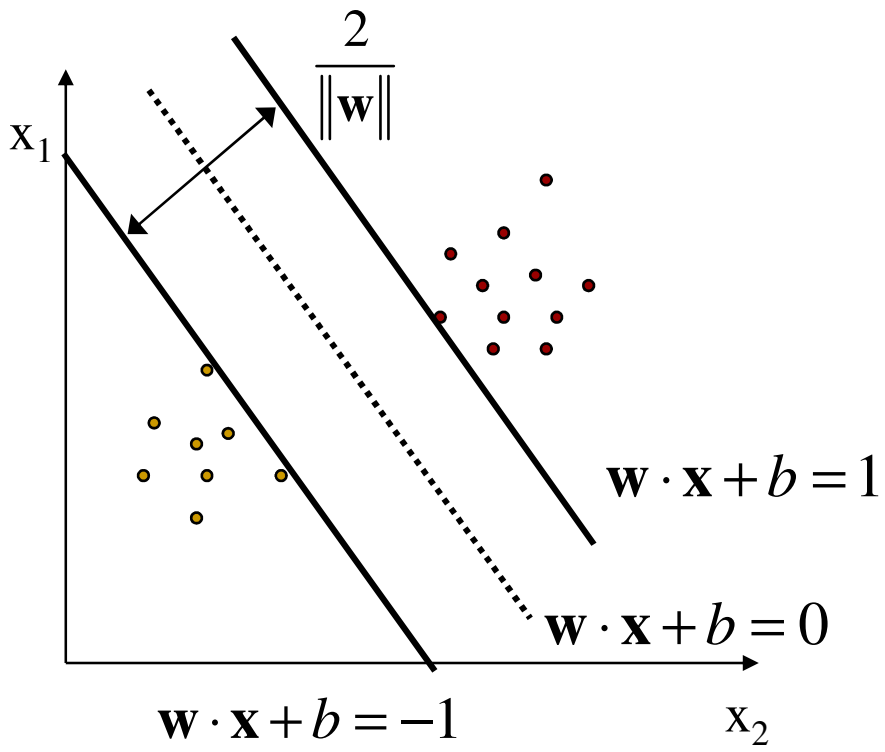
Šírka hranice medzi triedami je

$$M=(\mathbf{x}^{+}-\mathbf{x}^{-})\cdot\mathbf{n}$$

$$M=(\mathbf{x}^{+}-\mathbf{x}^{-})\cdot\frac{\mathbf{w}}{\|\mathbf{w}\|}=\frac{2}{\|\mathbf{w}\|}$$

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

Formálna definícia problému



Cieľom je maximalizovať hranicu pričom musia byť splnené podmienky správnej klasifikácie

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

$$s.t.$$

$$(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \ \forall \mathbf{x}_i \text{ z triedy 1}$$

$$(\mathbf{w} \cdot \mathbf{x}_i + b) \leq -1, \ \forall \mathbf{x}_i \text{ z triedy 2}$$

Formálna definícia problému

- Ak prvá triedu klasifikujeme ako hodnotu1 a druhú triedu ako hodnotu -1, môžno podmienky

$$(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1,\ \forall \mathbf{x}_i \text{ z triedy 1, t.j. } y_i = 1$$
$$(\mathbf{w}\cdot\mathbf{x}_i + b) \leq -1,\ \forall \mathbf{x}_i \text{ z triedy 2, t.j. } y_i = -1$$

- upraviť na

$$y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1,\ \forall \mathbf{x}_i$$

- Optimalizačný problém má teda tvar

$$\max \frac{2}{\|\mathbf{w}\|} \quad s.t.\ y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1,\ \forall \mathbf{x}_i$$

alebo

$$\min \frac{1}{2}\|\mathbf{w}\|^2 \quad s.t.\ y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1,\ \forall \mathbf{x}_i$$

Hard-Margin SVM

- Treba nájsť parametre ***w,b*** ktoré sú riešením

$$\min \frac{1}{2}\|\mathbf{w}\|^2$$

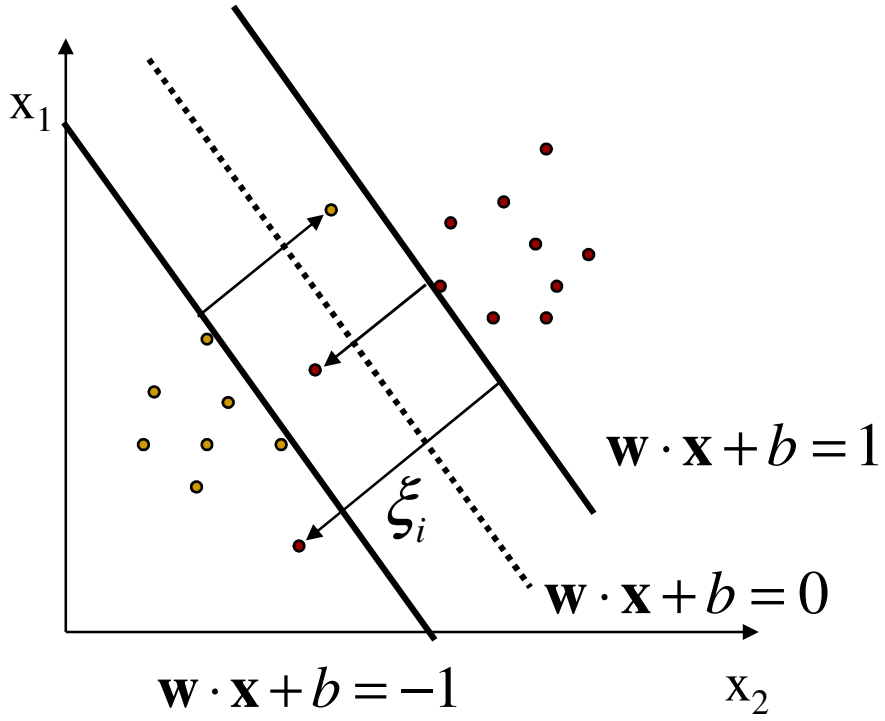
$$s.t.\ y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1,\ \ \forall \mathbf{x}_i$$

- Konvexný problém (kvadradická funkcia) garantuje jedinečné globálne minimum - **ak existuje!**
- Ak sú vstupné vzory (lineárne) neseparovateľné uvedená formulácia problému nenájde riešenie

Support Vector Machines

- Definujú optimálnu deliacu rovinu - **maximalizovanie hranice medzi triedami**
- Dokážu fungovať aj na lineárne neseparovateľných problémoch – **pomocou penalizácie za zlú klasifikáciu**
- Transformujú vstupy do “priestoru zaujímavých vlastností” (feature space) – **problém je preformulovaný tak aby v sebe transformáciu zahŕňal implicitne**

Lineárne neseparovateľné vstupné vzory



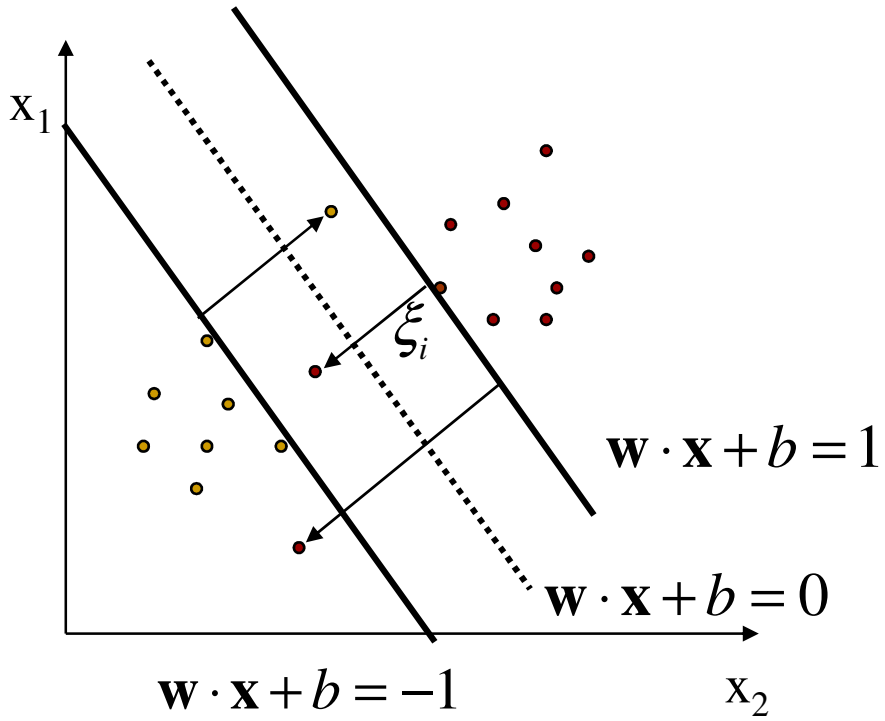
Zavedieme tolerančné premenné

$$\xi_i$$

Umožnia aby sa niektoré vstupné vzory nachádzali za klasifikačnými hranicami

Tieto vzory však budeme penalizovať

Preformulovanie optimalizačného problému



Obmedzenia majú tvar:

$$y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\geq 1-\xi_i,\ \forall\mathbf{x}_i$$
$$\xi_i\geq 0$$

Objektívna funkcia penalizuje zlé klasifikované vzory, ako aj tie ktoré sa nachádzajú v hraničnom pásme

$$\min\frac{1}{2}\|\mathbf{w}\|^2+C\sum_i\xi_i$$

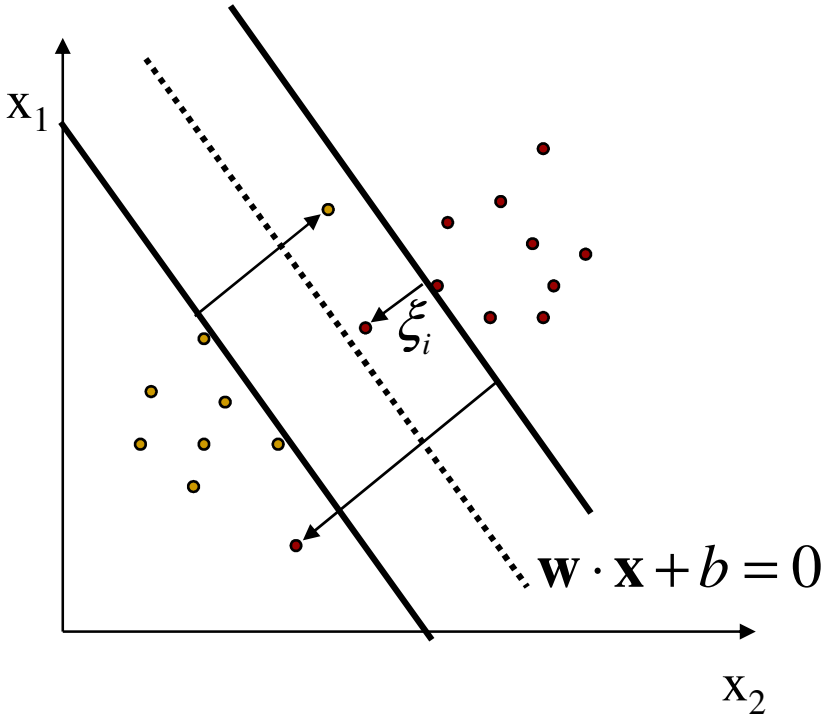
Parameter *C* dolaďuje vplyv nesprávane klasifikovaných vzorov a šírky hranice pri optimalizácii

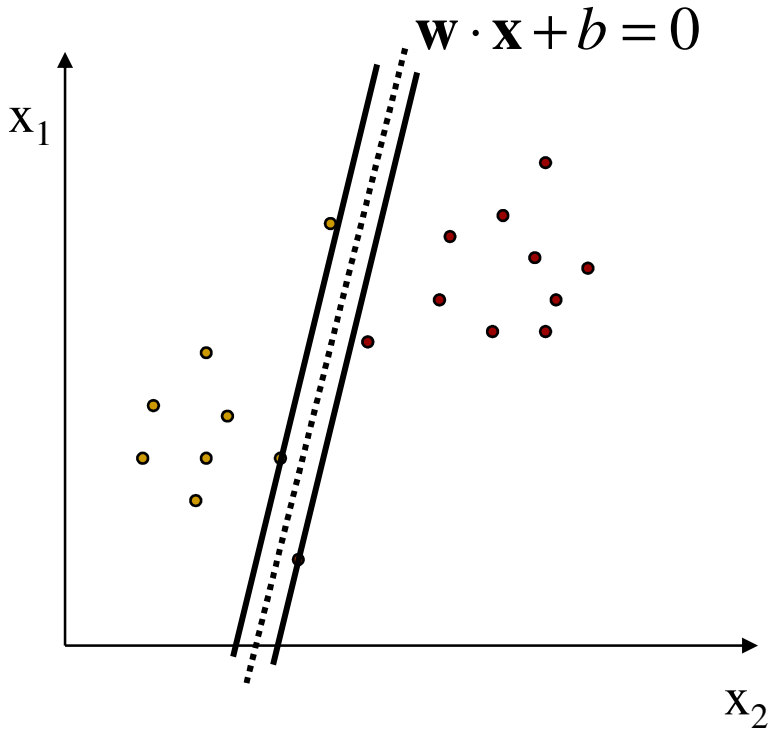
Soft-Margin SVM

$$\min \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \qquad y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i, \ \forall \mathbf{x}_i$$
$$\xi_i \geq 0$$

- Algoritmus sa snaží ponechať tolerančné premenné ξ_i na nulovej hodnote pričom súčasne maximalizuje oddelujúcu hranicu
- Algoritmus neminimalizuje počet nesprávne klasifikovaných vzorov (NP-complete problem) ale minimalizuje len sumu vzdialeností od rozhodovacích
- Väčšia hodnoda $C \to \infty$, spôsobí že riešenie sa bude približovať k hard-margin SVM

Soft vs. Hard Margin SVM





- Soft margin SVM
  - vždy existuje riešenie
  - sú robustnejšie v pripade zašumených vstupov

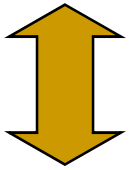
- Hard margin SVM
  - nemusí existovať riešenie
  - nie je potrebné odhadovať hodnotu parametra C

Hľadanie riešenia - optimalizácia

Kvadratické programovanie s lineárnymi obmedzeniami

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{s.t.} \quad y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1$$



Lagrangián

$$\text{minimize} \quad L_p(\mathbf{w}, b, \alpha_i) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 \right)$$

$$\text{s.t.} \quad \alpha_i \geq 0$$

Hľadanie riešenia - optimalizácia

$$\text{minimize } L_p(\mathbf{w}, b, \alpha_i) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \right)$$

$$\text{s.t.} \quad \alpha_i \geq 0$$

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

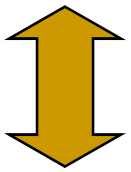
$$\frac{\partial L_p}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

Hľadanie riešenia - optimalizácia

$$\text{minimize } L_p(\mathbf{w}, b, \alpha_i) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \right)$$

$$\text{s.t.} \quad \alpha_i \geq 0$$

Lagrangián - Duálny Problém



$$\text{maximize } \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

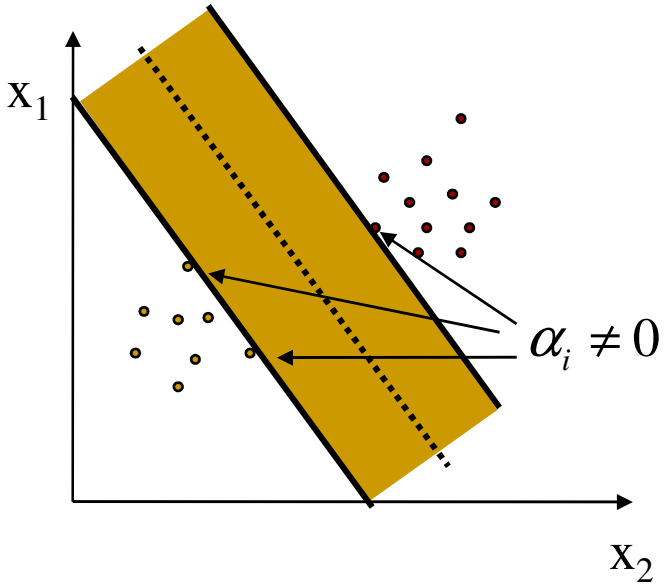
$$\text{s.t.} \quad \alpha_i \geq 0 \text{ , and } \sum_{i=1}^{n} \alpha_i y_i = 0$$

Hľadanie riešenia - optimalizácia

- Keďže musí platiť:

$$\alpha_i \left(y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \right) = 0$$

- **Iba podporné vektory majú** $\alpha_i \neq 0$



- Riešenie má tvar:

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = \sum_{i \in \mathrm{SV}} \alpha_i y_i \mathbf{x}_i$$

b sa da urcit z $y_i (\mathbf{w} \cdot \mathbf{x}_i + b) - 1 = 0,$

kde $\mathbf{x}_i$ je support vector

Algoritmy pre trénovanie SVM

- Lagrangeove SVM - LSVM
- Newtonova SVM – NSVM
- Sequential minimal optimization – SMO
- SVM light
- atď.

- Distribuované prístupy
  - Riešia problémy s veľkým počtom trénovacích vzorov

Hľadanie riešenia - optimalizácia

- Rozhodovacia funkcia má tvar:

$$g(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b = \sum_{i \in \mathrm{SV}} \alpha_i y_i \mathbf{x}_i \mathbf{x} + b$$

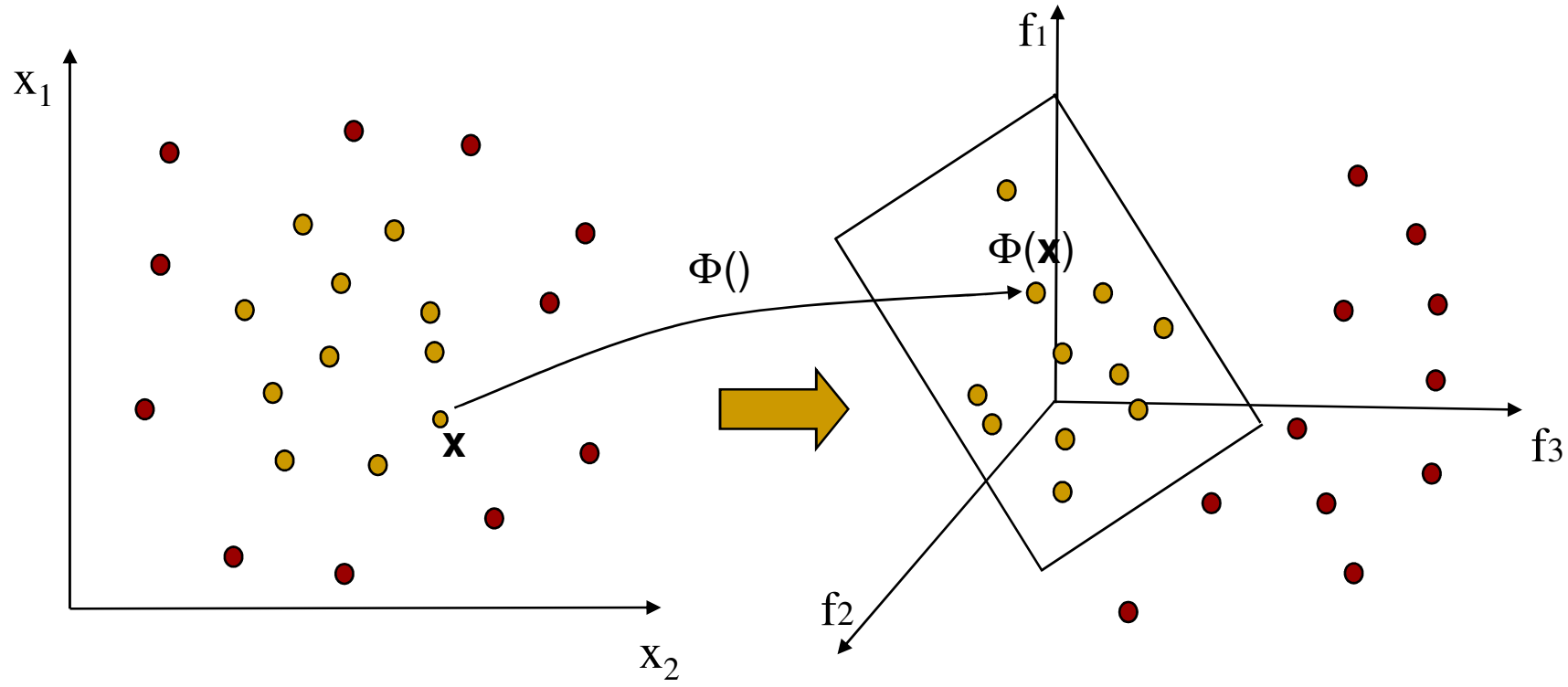
- Je založená na **skalárnom súčine** medzi testovaným bodom **x** a podpornými vektormi $\mathbf{x}_i$

- Hľadanie riešenia optimalizačného problému v sebe zahŕňa výpočet skalárneho súčinu medzi všetkými dvojicami vstupných vzorov $\mathbf{x}_i$ $\mathbf{x}_j$

Support Vector Machines

- Definujú optimálnu deliacu rovinu - **maximalizovanie hranice medzi triedami**
- Dokážu fungovať aj na lineárne neseparovateľných problémoch – **pomocou penalizácie za zlú klasifikáciu**
- Transformujú vstupy do “priestoru zaujímavých vlastností” (feature space) – **problém je preformulovaný tak aby v sebe transformáciu zahŕňal implicitne**

Nelineárne SVM – priestor zaujímavých vlastností



Funkcia Φ(**x**) transformuje vstupné vzory do “mnohorozmerného” priestoru zaujímavých vlastností

Kernel trick

- $\Phi(x_i) \cdot \Phi(x_j)$ predstavuje transformovanie vstupných vzorov do nového priestoru a následne vypočítanie skalárneho súčinu
- Ak dokážeme nájsť funkciu pre ktorú platí:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i).\Phi(\mathbf{x}_j)$$

- t.j. jej výsledok predstavuje skalárny súčin obrazov v mnohorozmernom priestore, nemusíme vykonávať explicitné transformovanie vstupov do mnohorozmerného priestoru.
- Klasifikácia nového vzoru sa dá totiž spraviť nasledovne:

$$\text{sgn}(\mathbf{w} \cdot \mathbf{x} + b) = \text{sgn}(\sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b)$$

kde b je riesenim $\alpha_j (y_j \sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j) + b - 1) = 0,$

pre lubovolne j kde $\alpha_j \neq 0$

Kernel trick

- Príklad:

$$\mathbf{x}_i = [x_{i1}, x_{i2}], \mathbf{x}_j = [x_{j1}, x_{j2}]$$

- Nech $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i \cdot \mathbf{x}_j)^2$

- Je potrebné ukázať

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i).\Phi(\mathbf{x}_j)$$

$$K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i \cdot \mathbf{x}_j)^2$$

$$= 1 + x_{i1}^2 x_{j1}^2 + 2x_{i1}x_{j1}x_{i2}x_{j2} + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}$$

$$= [1\ , x_{i1}^2, \sqrt{2}x_{i1}x_{i2}, x_{i2}^2, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}].[1\ , x_{j1}^2, \sqrt{2}x_{j1}x_{j2}, x_{j2}^2, \sqrt{2}x_{j1}, \sqrt{2}x_{j2}]$$

$$= \Phi(\mathbf{x}_i).\Phi(\mathbf{x}_j), \textit{kde}\ \Phi(\mathbf{x}) = [1\ , x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2]$$

Kernel trick

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^p$$

- Sa nazýva polynomyálny kernel stupňa *p*.
- Ak *p=2*, a vstupný vzor (vektor) má 7000 prvkov použitie kernela znamená vypočítať skalárny súčin so 7000 členmi a jeho následné umocnenie dvoma
- Ak by sme mali použiť explicitnú transformáciu do mnohorozmerného priestoru znamenalo by to výpočet približne 50,000,000 nových zaujímavých vlastnosti pre oba trénovacie vzory a následne počítať skalárny súčin uvedených vektorov
- Kenel trik teda umožňuje výrazným spôsobom znižovať výpočtovú náročnosť v porovnaní s explicitnou transformáciou

Mercerova podmienka

- Nie každá symetrická funkcia $K(x,z)$ predstavuje kernel, t.j. Nemusí existovať zodpovedaujúca transformácia $\Phi(x)$
- Duálna formulácia SVM vyžaduje výpočet tzv. Gramovej matice $G_{ij} = K(x_i , x_j)$ ktorá obsahuje hodnoty $K(x_i, x_j)$ pre každú dvojicu trénovacích vzorov
- Transformácia $\Phi(x)$ do priestor zaujímavých vlastností existuje keď je matica G semi pozitivne definitná (Mercerova podmienka)

Príklady kernelových funkcií

- Príklady kernelových funkcií:
  - Linearny kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
  - Polynomiálny kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^p$
  - Gaussovský (Radial-Basis Function (RBF)) kernel:
$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{2\sigma^2}\right)$$
  - Sigmoidálny kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta_0 \mathbf{x}_i^T \mathbf{x}_j + \beta_1)$
- Funkcie ktoré splnujú tzv. *Mercerovu podmienku* môžu byť kernelové funkcie.

SVM

- Voľba kernelovej funkcie
  - Gaussovský, resp. polynomiálny kernel je prvá voľba
  - Ak nevyhovuje, je nutné nájsť iné pokročilé kernely
  - Spravidla je nutná znalosť problémovej domény
- Voľba parametrov kernela
  - Napr. σ pri Gaussovskom kerneli
  - σ predstavuje vzdialenosť medzi najbližšími vstupnými vzormi s rozdielnou triedou klasifikácie
  - Inou možnosťou je využiť validačné množiny, resp. cross validáciu
- Optimalizácia – Hard margin v.s. Soft margin
  - Spravidla viacero sérií experimentov s rôznymi parametrami

This slide is courtesy of *www.iro.umontreal.ca/~pift6080/documents/papers/**svm**_tutorial.**ppt***

Iné druhy Kernelových metód

- SVM pre regressiu (SVR)
- SVM pre klasterizáciu
- Kernel PCA
- Aplikácie:
  - Detekcia a rozpoznávanie tvárí v obrázku
  - Spracovanie textu – klasifikácia dokumentov / filtrovanie emailov
  - Bioinformatické aplikácie
    - hľadanie kódujúcej sekvencie v DNA
    - potvrdenie diagnózy

Klasifikácia pri viacerých triedach

- Jeden voči všetkým (One-versus-all)
  - *n* binárnych klasifikátorov, každý pre jednu triedu vs všetky ostatné triedy.
  - Vyberie sa tireda ktorá má najväčšiu pravdepodobnosť
- Jeden voči jednému (One-versus-one)
  - *n(n-1)/2* klasifikátorov, každý rozlišuje medzi jednou dvojicou tried
  - Viacero stratégií pre výber výslednej tiredy na základe výstupu binárnych výstupov SVM
- MultiClass SVMs
  - Zobšeobecnenie formalizmu SVM pre viacero tried

Porovnanie s neuronovými sietami

Neurónové siete

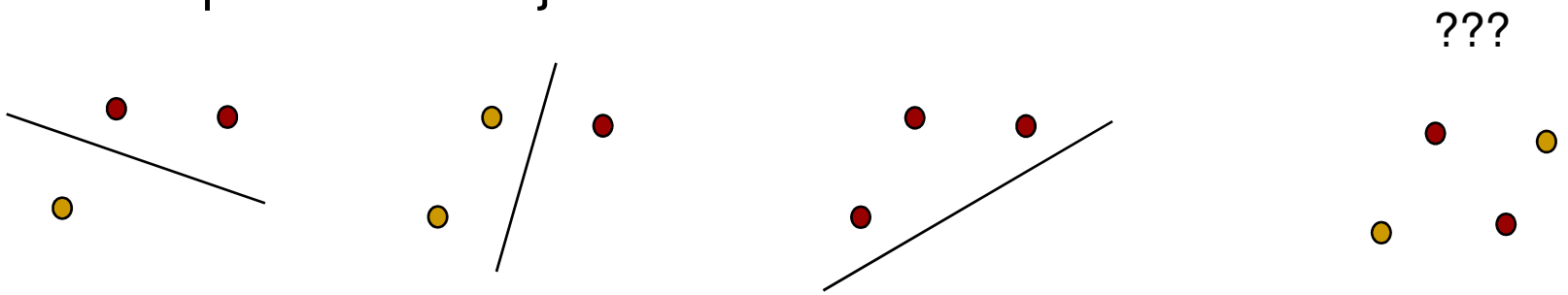
- Skrytá vrstva transformuje do nízko rozmerného feature space
- Prehľadavany priestor váh ma viacero lokálnych miním
- Trénovanie je výpočtovo náročné
- Klasifikácie je extrémne efektívna
- Vyžaduje stanoviť počet skrytých neurónov a vrstiev

SVM

- Kernelové funkcie transformujú vstupy do mnohorozmerných priestorov
- Prehľadávaný priestor má jedinečné minimum
- Trénovanie aj klasifikácia je extrémne efektívne
- Je potrebné zvoliť Kernelovú funkciu a parameter C
- Robustný prístup avšak citlivý na zle označené vstupy

Prečo sú SVM zovšeobecňujú?

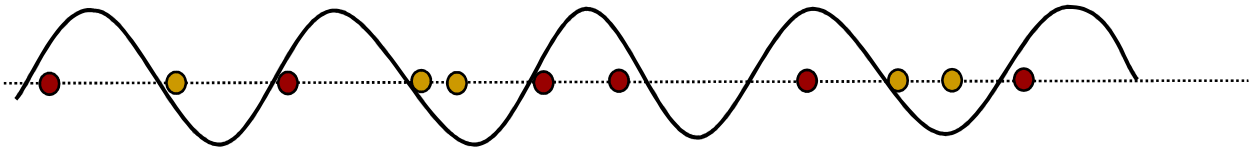
- VC dimenzia
  - Najvyšší počet vstupných vzorov, ešte ktoré dokáže model správne klasifikovať
  - Pre priamku v R^2 je VC rovná 3



  - Pre hyperrovinu v R^n je VC rovná n+1
  - VC meria zložitosť zvolenej triedy rozhodovacích funkcií
  - Nesúvisí ale s počtom jej parametrov!!

Prečo sú SVM zovšeobecňujú?

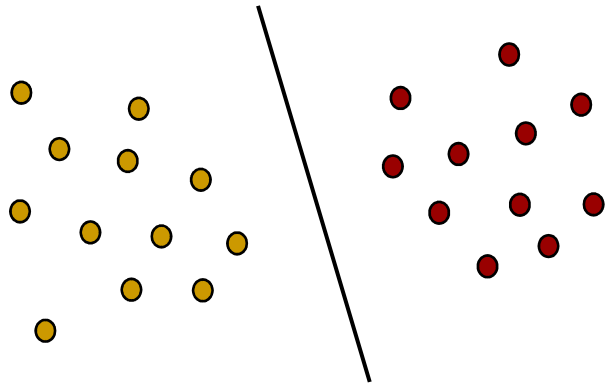
- VC dimenzia
  - Nesúvisí ale s počtom jej parametrov!!
  - Napr. f(x,w)=sign(sin(w.x)), w∈ R x∈ R



  - Pre ľubovolný počet vstupných vzorov vždy existuje hodnota parametera w, ktorá správne klasifikuje
  - VC dimenzia tejto triedy funkcií je ∞

Prečo sú SVM zovšeobecňujú?

- VC dimenzia
  Pre priamku v R^2 je VC rovná 3



- Ak je počet vstupných vzorov >> VC dimenzia modelu, získavame istotu, že model je vhodný na klasifikáciu

- Pri SVM je VC dimenzia závislá len od šírky oddeľujúcej hranice a nezávisí od rozmeru vstupného vektora, resp. počtu parametrov.

Záver

- Predstavujú veľmi dôležitý objav z oblasti strojového učenia
- Nájdenie hyperplochy ktorá zabezpečí najväčšie oddelenie dvoch tried vstupných vzorov
- Kernel trick – umožní efektívne pracovať v mnohorozmernom priestore zaujímavých vlastností

Zdroje

Prezentácie

- *www.iro.umontreal.ca/~pift6080/documents/papers/svm_tutorial.ppt*
- www.cs.cmu.edu/~awm/tutorials
- www.site.uottawa.ca/~stan/csi5387/SVM-appl.pdf
- http://www.cadlm.fr/%5Cworkshop%5Cdoc%5Cproceedings/Kxen.ppt

Web

- www.kernel-machines.org
- www.kernel-methods.net
- www.support-vector.net
- www.support-vector-machines.org