



# Teorie kognitivních systémů

## 7 Support Vector Machines

- Úvod k SVM
- Cíl optimalizace (alternativní pohled na logistickou regresi)
- Matematický model SVM
- Hypotéza s bezpečnostním faktorem
- Jádra (Kernels)
- Aplikace





# Support Vector Machines (SVM)

## Úvod

**SVM** – metoda strojového učení, která v klasifikační úloze slouží k **nalezení optimální rozdělující nadroviny oddělující trénovací data** v prostoru příznaků...

**Optimální nadrovina** – body projekce trénovacích dat leží na jejích opačných stranách, tj. v poloprostorech, které tato nadrovina odděluje, a minimum vzdáleností bodů od této nadroviny je co největší (tj. po obou stranách této nadroviny je co nejširší pás bez bodů).

K popisu takové nadroviny stačí pouze nejbližší body, kterých je obvykle málo – tzv. **podpůrné vektory (Support Vectors)**

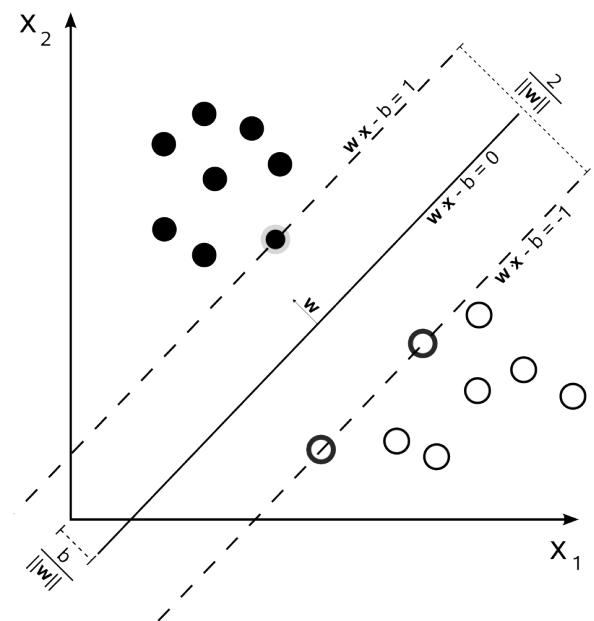
SVM je binární klasifikační technika, rozdělující nadrovina je v prostoru příznaků lineární funkcí...



# Support Vector Machines (SVM)

## Úvod

- velmi populární a široce užívaná technika učení s učitelem
- mnohdy poskytují „čistší“ a výkonnější způsob nalezení rozdělující nadplochy než regrese či např. neuronové sítě
- známé také pod názvem **klasifikátory s širokým okrajem** (*Large Margin Classifier*)





# Cíl optimalizace

## Alternativní pohled na logistickou regresi

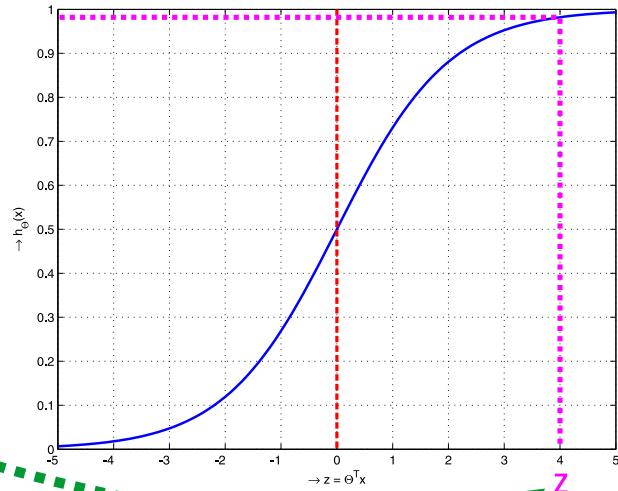
Modifikace logistické regrese poskytuje základní aparát SVM:

- ve vzorci hypotézy logistické regrese

$$h_{\Theta}(x) = \frac{1}{1 + e^{-\Theta^T x}}$$

přeznačíme  $\Theta^T x = z$

- jestliže  $y = 1$ , požadujeme, aby  $h_{\Theta}(x) \approx 1 \rightarrow \Theta^T x \gg 0$
- jestliže  $y = 0$ , požadujeme, aby  $h_{\Theta}(x) \approx 0 \rightarrow \Theta^T x \ll 0$





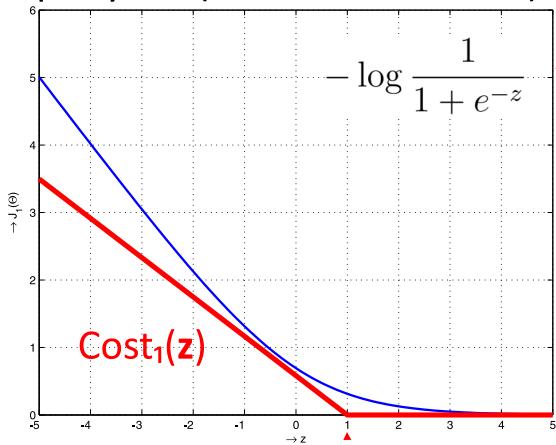
# Cíl optimalizace

## Modifikace cenové funkce

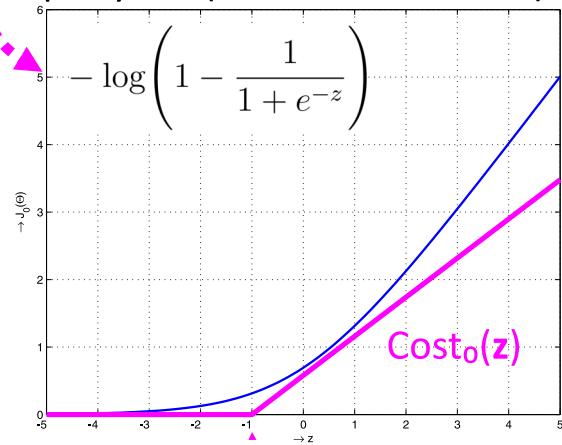
Vzorek  $(x, y)$  z trénovací množiny pak k celkové hodnotě  $J(\Theta)$ , kterou minimalizujeme, přispívá hodnotou:

$$\begin{aligned}
 & - \left( y \log h_{\Theta}(\mathbf{x}) + (1 - y) \log(1 - h_{\Theta}(\mathbf{x})) \right) \\
 &= -y \log \frac{1}{1 + e^{-\Theta^T x}} - (1 - y) \log \left( 1 - \frac{1}{1 + e^{-\Theta^T x}} \right)
 \end{aligned}$$

pro  $y = 1$  (chtěme  $\Theta^T x \gg 0$ ):



pro  $y = 0$  (chtěme  $\Theta^T x \ll 0$ ):



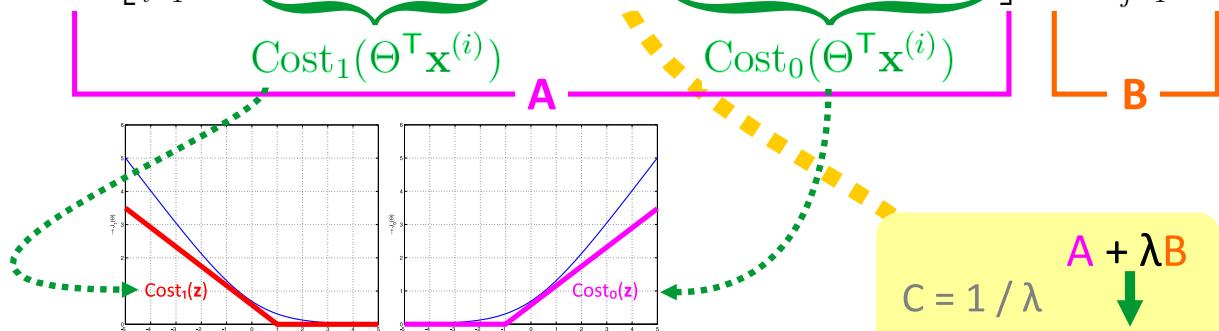


# Support Vector Machine

## Matematický model

Logistická regrese:

$$\min_{\Theta} \frac{1}{m} \left[ \sum_{i=1}^m y^{(i)} \underbrace{\left( -\log h_{\Theta}(\mathbf{x}^{(i)}) \right)}_{\text{Cost}_1(\Theta^T \mathbf{x}^{(i)})} + (1-y^{(i)}) \underbrace{\left( -\log(1-h_{\Theta}(\mathbf{x}^{(i)})) \right)}_{\text{Cost}_0(\Theta^T \mathbf{x}^{(i)})} \right] + \frac{\lambda}{2m} \sum_{j=1}^n \Theta_j^2$$



Support Vector Machine:

$$\min_{\Theta} C \sum_{i=1}^m \left[ y^{(i)} \text{Cost}_1(\Theta^T \mathbf{x}^{(i)}) + (1-y^{(i)}) \text{Cost}_0(\Theta^T \mathbf{x}^{(i)}) \right] + \frac{1}{2} \sum_{j=1}^n \Theta_j^2$$



# Support Vector Machine

## Tvar hypotézy

Na rozdíl od logistické regrese **hypotéza používaná SVM nevyjadřuje pravděpodobnost**, ale přímo příslušnost klasifikovaného vzorku k třídě 1 nebo 0.

Optimalizací výrazu

$$\min_{\Theta} C \sum_{i=1}^m \left[ y^{(i)} \text{Cost}_1(\Theta^T \mathbf{x}^{(i)}) + (1 - y^{(i)}) \text{Cost}_0(\Theta^T \mathbf{x}^{(i)}) \right] + \frac{1}{2} \sum_{j=1}^n \Theta_j^2$$

získáme hodnoty parametrů  $\Theta_i$ , které určují výpočet hypotézy.

**Hypotéza SVM:**

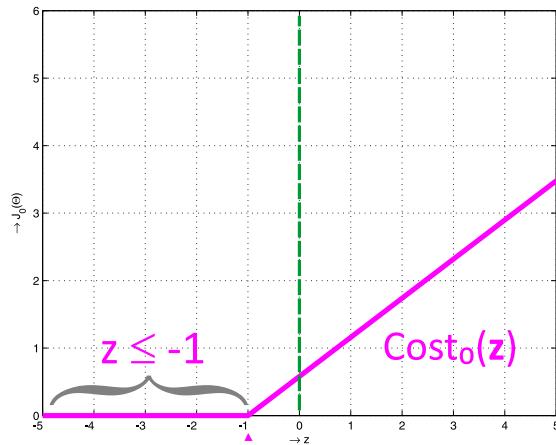
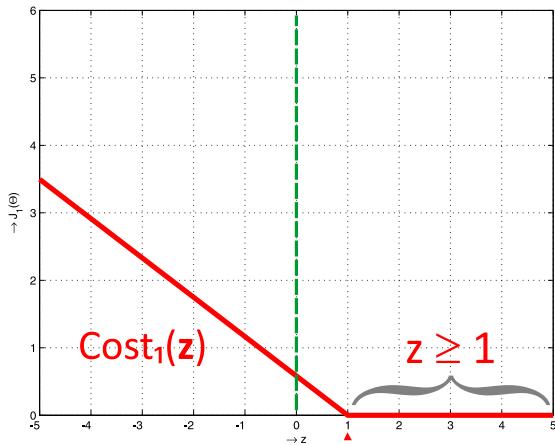
$$h_{\Theta}(\mathbf{x}) = \begin{cases} 1 & \text{tehdy a jen tehdy, když } \Theta^T \mathbf{x} \geq 0 \\ 0 & \text{jinak} \end{cases}$$



# Support Vector Machine

## Bezpečnostní faktor

$$\min_{\Theta} C \sum_{i=1}^m \left[ y^{(i)} \text{Cost}_1(\Theta^\top \mathbf{x}^{(i)}) + (1 - y^{(i)}) \text{Cost}_0(\Theta^\top \mathbf{x}^{(i)}) \right] + \frac{1}{2} \sum_{j=1}^n \Theta_j^2$$



↓ tvar cenové funkce vytváří  
„bezpečnostní“ zónu

Je-li  $y = 1$ , chceme, aby  $\Theta^\top \mathbf{x} \geq 1$  (nikoliv jen  $\geq 0$ , jako u LogR).  
Je-li  $y = 0$ , chceme, aby  $\Theta^\top \mathbf{x} \leq -1$  (nikoliv jen  $\leq 0$ ).



# Support Vector Machine

## Rozhodovací hranice

Uvažujme regularizační faktor  $C$  velmi vysoký, např. 100 000:

$$\min_{\Theta} C \sum_{i=1}^m \left[ y^{(i)} \text{Cost}_1(\Theta^T \mathbf{x}^{(i)}) + (1 - y^{(i)}) \text{Cost}_0(\Theta^T \mathbf{x}^{(i)}) \right] + \frac{1}{2} \sum_{j=1}^n \Theta_j^2$$

A → 0
A → 0

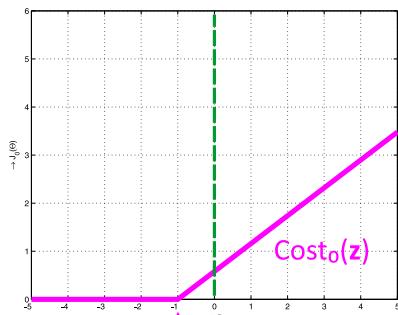
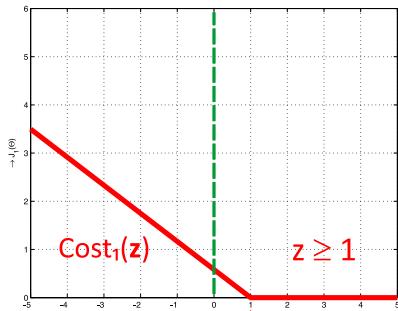
taková volba  $C$  způsobí,  
že výsledkem optimali-  
zace bude hodnota vý-  
razu  $A$  velmi blízká 0...

pro  $y^{(i)} = 1$ :  $\Theta^T \mathbf{x}^{(i)} \geq 1$ ,

pro  $y^{(i)} = 0$ :  $\Theta^T \mathbf{x}^{(i)} \geq -1$

optimalizujeme  $\min_{\Theta} C \cdot 0 + \frac{1}{2} \sum_{j=1}^n \Theta_j^2$

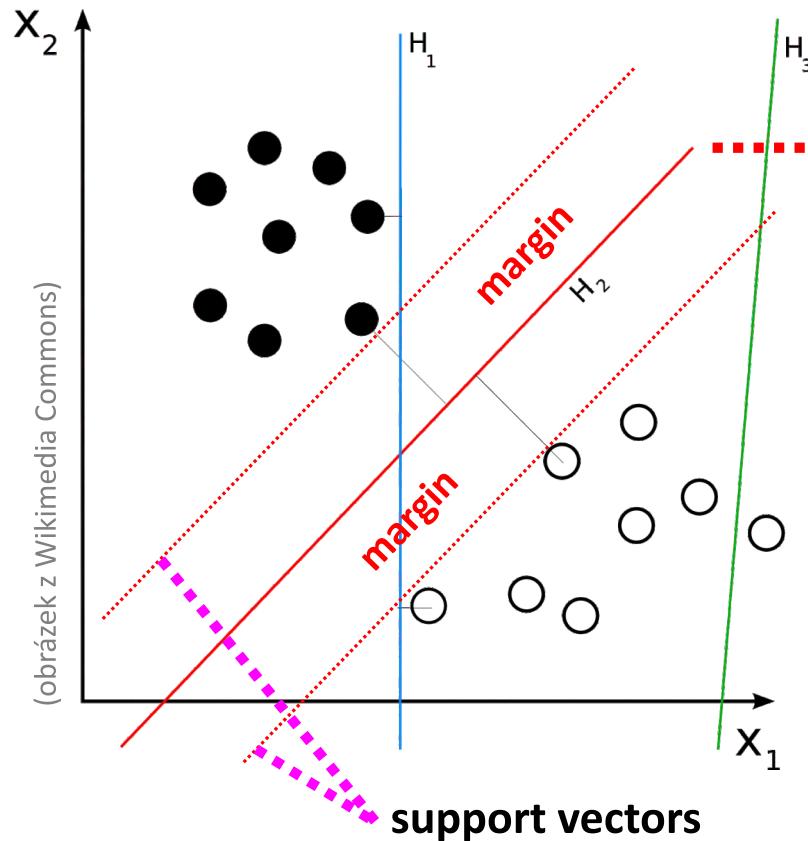
Výsledkem je velmi zajímavý  
tvar rozhodovací hranice...





# Rozhodovací hranice SVM

## Případ lineárně separabilních dat



SVM vybere tuto rozhodovací hranici ( $H_2$ )

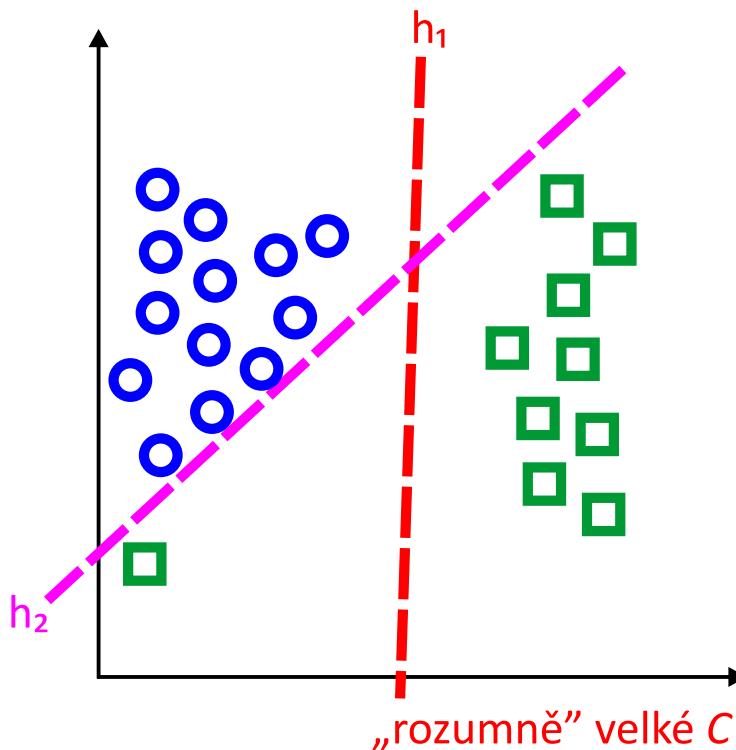
– optimalizace preferuje takový tvar hypotézy, že okraj (Margin) je největší možný...

→ **Large Margin Classifier**



# Rozhodovací hranice SVM

## Případ lineárně téměř neseparabilních dat

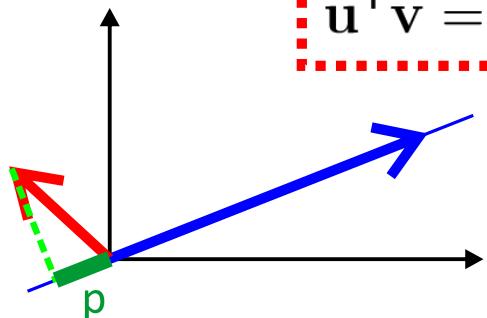
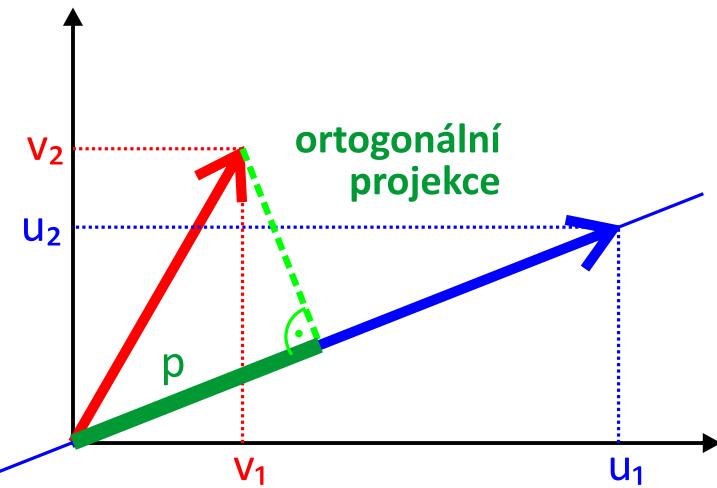


je-li regularizační faktor  $C$  velmi vysoký a objeví-li se vzorek v extrémní pozici, optimalizace „vybere“ hypotézu  $h_2$  namísto vhodnější  $h_1$



# Matematika v pozadí SVM

## Skalární součin vektorů



$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

Co je **skalární součin**  $\mathbf{u}^T \mathbf{v}$   
a jaký má geom. význam?

$$\|\mathbf{u}\| = \sqrt{u_1^2 + u_2^2} \in \mathbb{R}$$

$$\boxed{\mathbf{u}^T \mathbf{v} = \mathbf{v}^T \mathbf{u} = p \cdot \|\mathbf{u}\| = u_1 v_1 + u_2 v_2}$$

norma ("délka") vektoru  
délka projekce v na u  
 $(= \| \mathbf{v} \| \cos \alpha)$



# Matematika v pozadí SVM

## Odvození tvaru rozhodovací hranice

Cíl optimalizace SVM:

$$\min_{\Theta} \frac{1}{2} \sum_{j=1}^n \Theta_j^2 \quad \left| \begin{array}{ll} \Theta^T x^{(i)} \geq 1 & \text{když } y^{(i)} = 1 \\ \Theta^T x^{(i)} \leq -1 & \text{když } y^{(i)} = 0 \end{array} \right.$$

$\|\Theta\|$

zjednodušme:  $\Theta_0 = 0, n = 2$

$$= \frac{1}{2} (\Theta_1^2 + \Theta_2^2) = \frac{1}{2} \left( \sqrt{\Theta_1^2 + \Theta_2^2} \right)^2$$

$$= \min_{\Theta} \frac{1}{2} \|\Theta\|^2$$



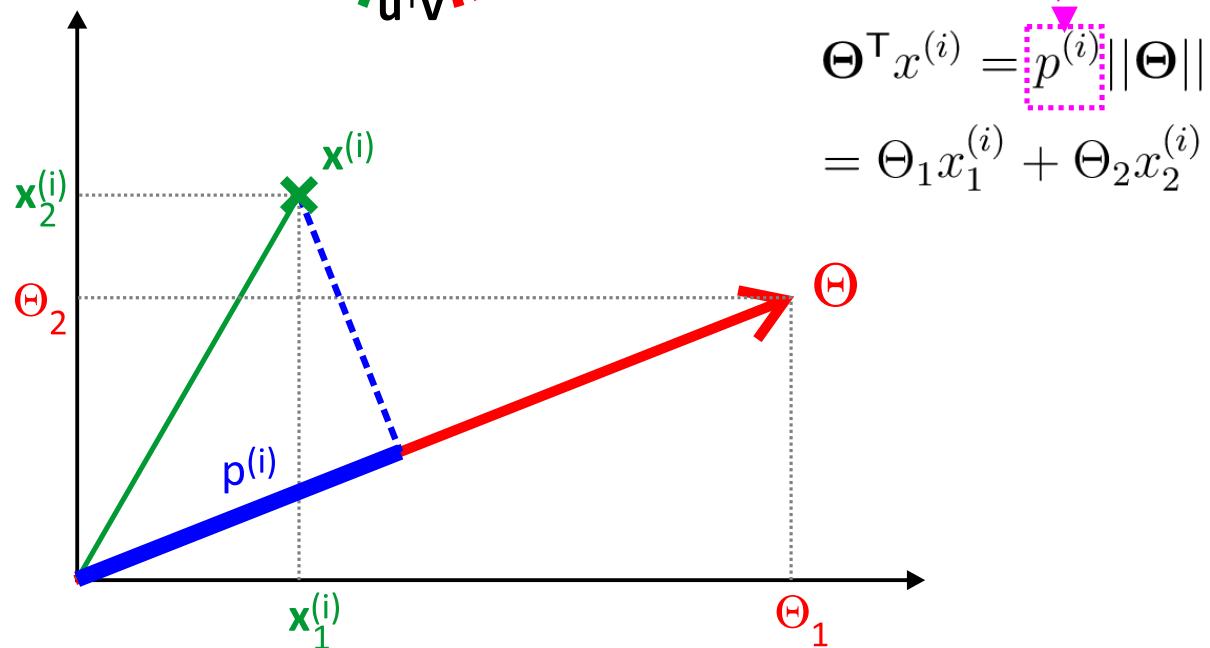
cílem optimalizace SVM  
tedy je minimalizace kvadratické normy vektoru  
parametrů  $\Theta_i$



# Matematika v pozadí SVM

## Odvození tvaru rozhodovací hranice

$$\min_{\Theta} \frac{1}{2} \sum_{j=1}^n \Theta_j^2 \quad \left| \begin{array}{ll} \Theta^T x^{(i)} \geq 1 & \text{když } y^{(i)} = 1 \\ \Theta^T x^{(i)} \leq -1 & \text{když } y^{(i)} = 0 \end{array} \right.$$





# Matematika v pozadí SVM

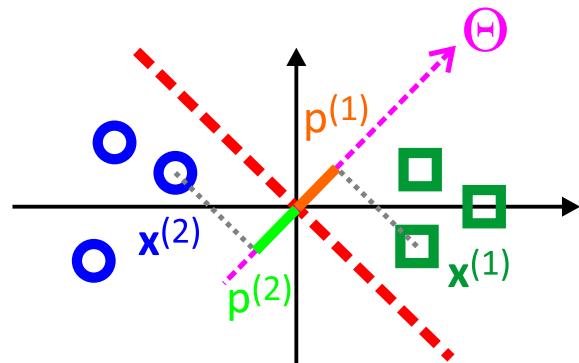
## Odvození tvaru rozhodovací hranice

$$\min_{\Theta} \frac{1}{2} \sum_{j=1}^n \Theta_j^2 = \min_{\Theta} \frac{1}{2} \|\Theta\|^2 \quad \left| \begin{array}{ll} p^{(i)} \|\Theta\| \geq 1 & \text{když } y^{(i)} = 1 \\ p^{(i)} \|\Theta\| \leq -1 & \text{když } y^{(i)} = 0 \end{array} \right.$$

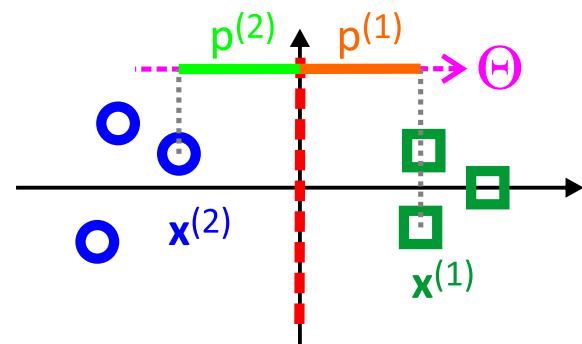
$p^{(i)}$  je projekce  $x^{(i)}$  na vektor  $\Theta$

Stále platí zjednodušení, že

$\Theta_0 = 0$  (matem. v pořádku – hranice prochází bodem  $[0, 0]$ ):



$p^{(1)} \|\Theta\| \geq 1 \Rightarrow \|\Theta\| \text{ velká}$   
 $p^{(2)} \|\Theta\| \leq -1 \Rightarrow \|\Theta\| \text{ velká}$

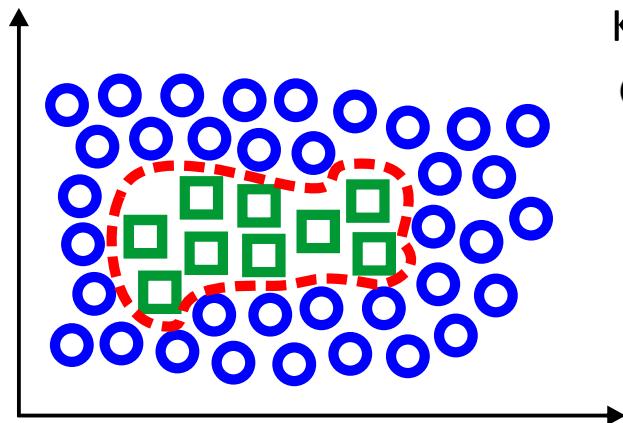


$p^{(1)} \|\Theta\| \geq 1 \Rightarrow \|\Theta\| \text{ menší}$   
 $p^{(2)} \|\Theta\| \leq -1 \Rightarrow \|\Theta\| \text{ menší}$



# Nelineární rozhodovací hranice

## Výběr příznaků



Klasifikátor predikuje  $y = 1$ , když  
 $\Theta_0 + \Theta_1 x_1 + \Theta_2 x_2 + \Theta_3 x_1 x_2 +$   
 $+ \Theta_4 x_1^2 + \Theta_5 x_2^2 + \dots \geq 0$

– tj. hypotéza má tvar

$$h_{\Theta}(\mathbf{x}) = \begin{cases} 1 & \Leftrightarrow \Theta_0 + \Theta_1 x_1 + \dots \geq 0 \\ 0 & \text{v opačném případě} \end{cases}$$

Zjednodušme zápis substitucí:

$$\Theta_0 + \Theta_1 f_1 + \Theta_2 f_2 + \Theta_3 f_3 + \Theta_4 f_4 + \Theta_5 f_5 + \dots$$

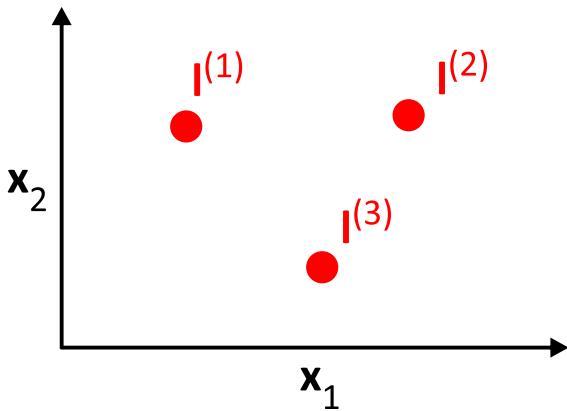
$$f_1 = x_1, f_2 = x_2, f_3 = x_1 x_2, f_4 = x_1^2, f_5 = x_2^2 \dots$$

Existuje jiný / lepší výběr příznaků  $f_1, f_2, f_3, \dots, f_n$ ?



# Jádra (Kernels)

## Lepší volba příznaků



V příznakovém prostoru **vybere-**  
**me** (např. ručně?) 3 body, tzv.  
**landmarks**  $l^{(1)}, l^{(2)}, l^{(3)}$ .

Příznaky vzorku  $x$  vypočteme  
jako funkci vzdálenosti k bodům  
 $l^{(1)}, l^{(2)}, l^{(3)}$ .

Pro  $x$

$$\left\{ \begin{array}{l} f_1 = \text{podobnost}(x, l^{(1)}) = e^{\left(-\frac{\|x-l^{(1)}\|^2}{2\sigma^2}\right)} \\ f_2 = \text{podobnost}(x, l^{(2)}) = e^{\left(-\frac{\|x-l^{(2)}\|^2}{2\sigma^2}\right)} \\ f_3 = \text{podobnost}(x, l^{(3)}) = e^{\left(-\frac{\|x-l^{(3)}\|^2}{2\sigma^2}\right)} \end{array} \right.$$

**kernel  $k(x, l^{(i)})$  (gaussovský)**



# Jádra (Kernels)

## Vyjádření podobnosti

$$f_1 = \text{podobnost}(\mathbf{x}, \mathbf{l}^{(1)}) = e^{\left(-\frac{\|\mathbf{x}-\mathbf{l}^{(1)}\|^2}{2\sigma^2}\right)} = e^{\left(-\frac{\sum_{j=1}^n (x_j - l_j^{(1)})^2}{2\sigma^2}\right)}$$

Je-li  $\mathbf{x}$  blízko  $\mathbf{l}^{(1)}$   $\mathbf{x} \approx \mathbf{l}^{(1)}$  :  $f_1 \approx e^{\left(-\frac{0^2}{2\sigma^2}\right)} \approx 1$

Naopak je-li  $\mathbf{x}$  daleko od  $\mathbf{l}^{(1)}$  :  $f_1 \approx e^{\left(-\frac{\text{velké číslo}^2}{2\sigma^2}\right)} \approx 0$

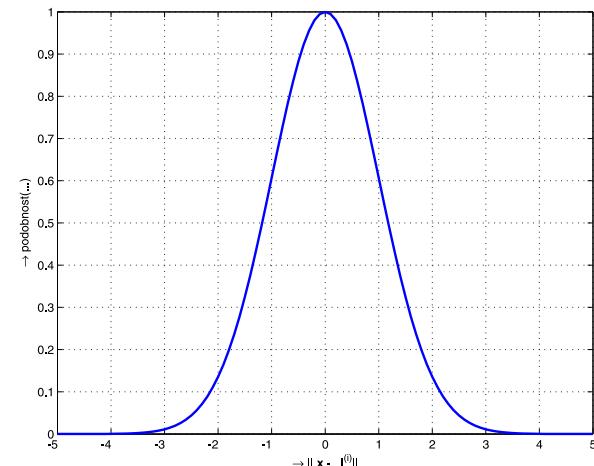
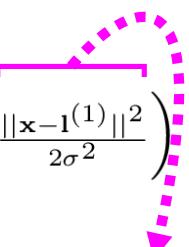
Výpočtem podobnosti vzorku  $\mathbf{x}$  s landmarky získáme nové příznaky:

$$\mathbf{l}^{(1)} \rightarrow f_1$$

$$\mathbf{l}^{(2)} \rightarrow f_2$$

...

$$\mathbf{l}^{(k)} \rightarrow f_k$$



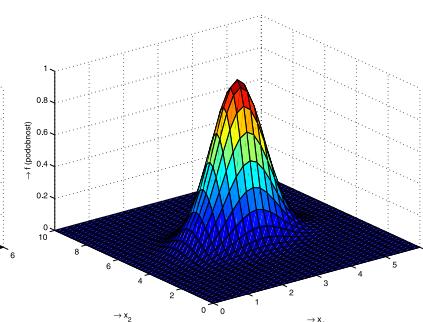
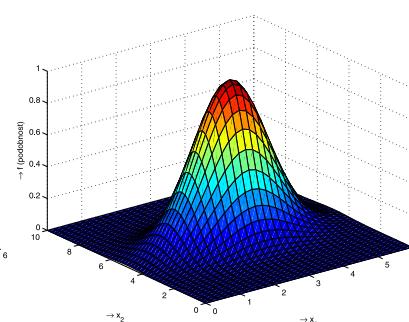
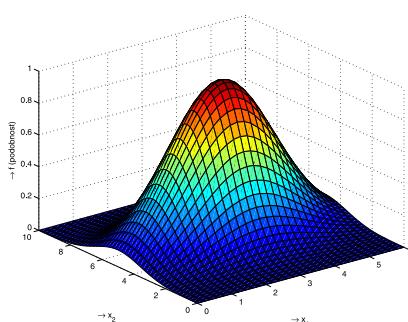
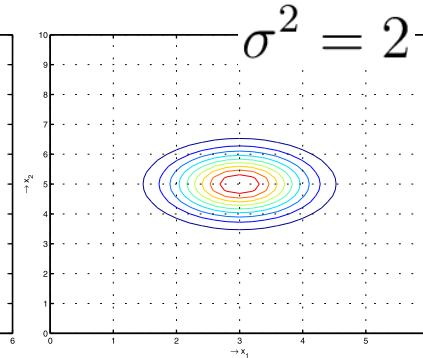
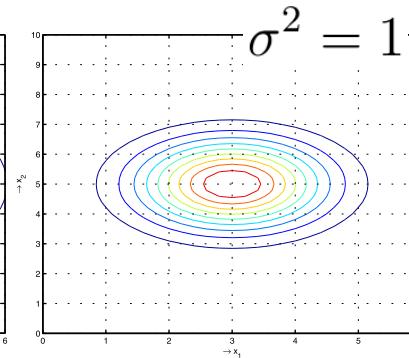
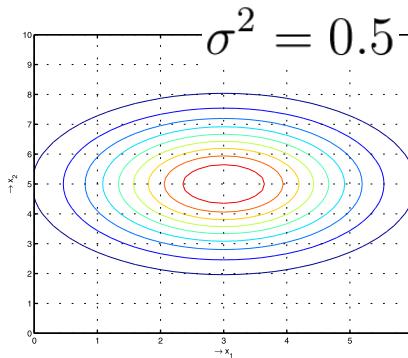


# Jádra (Kernels)

## Vyjádření podobnosti – příklad

$$\mathbf{l}^{(1)} = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$$

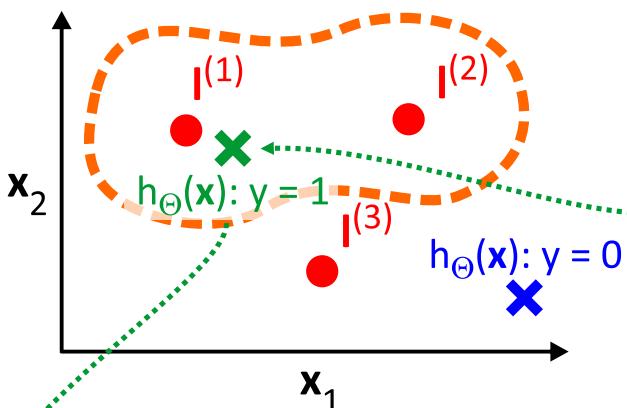
$$f_1 = \exp\left(-\frac{\|\mathbf{x} - \mathbf{l}^{(1)}\|^2}{2\sigma^2}\right)$$





# Jádra (Kernels)

## Tvar hypotézy a rozhodovací hranice



Hypotéza predikuje  $y = 1$ , je-li:

$$\Theta_0 + \Theta_1 f_1 + \Theta_2 f_2 + \Theta_3 f_3 \geq 0$$

$x$  – klasifikovaný vzorek s příznaky (podobnost landmarků)  
 $f_1 \leftarrow 1, f_2 \approx 0, f_3 \approx 0$

Trénováním získáme např. takovéto nastavení parametrů hypotézy:  $\Theta_0 = -0.5, \Theta_1 = 1, \Theta_2 = 1, \Theta_3 = 0$

Dosazením hodnot příznaků do hypotézy dostaneme:

$$\Theta_0 + \Theta_1 \times 1 + \Theta_2 \times 0 + \Theta_3 \times 0 = -0.5 + 1 = 0.5 \geq 0$$

$h_\Theta(x): y = 1$

$$f_1, f_2, f_3 \approx 0 : \Theta_0 + \Theta_1 f_1 + \dots \approx -0.5 \leq 0$$

$h_\Theta(x): y = 0$

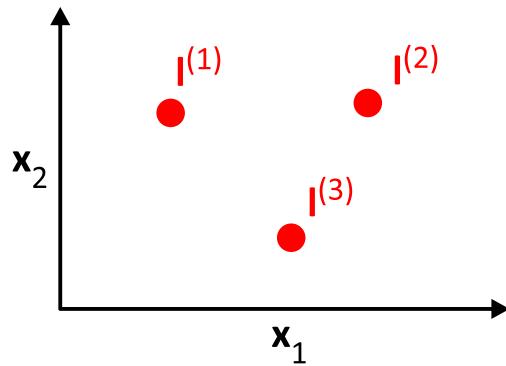




# Jádra (Kernels)

## Problémy nasazení SVM s jádry

- jak vybrat v příznakovém prostoru landmarky?



- jak zvolit míru podobnosti pro výpočet příznaků?
- programovat celý algoritmus?



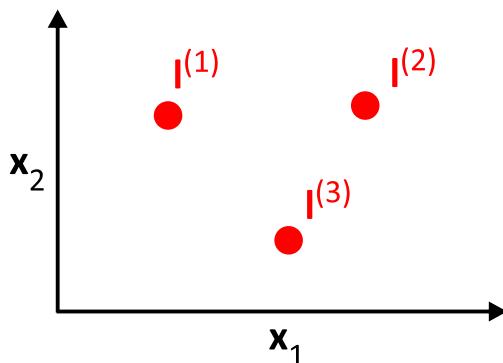
**Nešlo by techniku jader použít i u jiných klasifikačních algoritmů, např. u logistické regrese?**

- **ano, šlo**, ale bylo by to výpočetně velmi náročné...
- jádra fungují nejlépe ve spojení s aparátem SVM



# Výběr landmarků

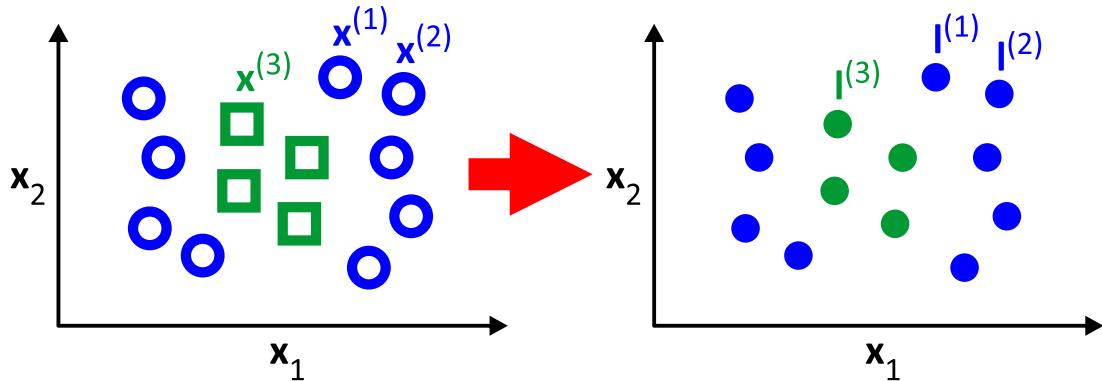
## Funkce trénovací množiny u SVM s jádry



Je-li dán vzorek  $\mathbf{x}$ , pak

$$\begin{aligned}f_i &= \text{podobnost}(\mathbf{x}, \mathbf{l}^{(i)}) \\&= \exp\left(-\frac{\|\mathbf{x} - \mathbf{l}^{(i)}\|^2}{2\sigma^2}\right)\end{aligned}$$

$h_{\Theta}(\mathbf{x})$  predikuje  $y = 1 \Leftrightarrow \Theta_0 + \Theta_1 f_1 + \Theta_2 f_2 + \Theta_3 f_3 \geq 0$   
Jak získat pozice  $\mathbf{l}^{(1)}, \mathbf{l}^{(2)}, \mathbf{l}^{(3)}, \dots$  ?





# Výběr landmarků

## Funkce trénovací množiny u SVM s jádry

Dána trénovací množina  $(\mathbf{x}^{(1)}, y^{(1)}) , (\mathbf{x}^{(2)}, y^{(2)}) , \dots , (\mathbf{x}^{(m)}, y^{(m)})$   
vybereme  $\mathbf{l}^{(1)} = \mathbf{x}^{(1)}, \mathbf{l}^{(2)} = \mathbf{x}^{(2)}, \dots, \mathbf{l}^{(m)} = \mathbf{x}^{(m)}$

Pak pro každý vzorek  $\mathbf{x}$  lze spočítat příznakový vektor  $\mathbf{f}$ :

$$\begin{aligned} f_1 &= \text{podobnost}(\mathbf{x}, \mathbf{l}^{(1)}) \\ f_2 &= \text{podobnost}(\mathbf{x}, \mathbf{l}^{(2)}) \\ \dots & \end{aligned}$$

$$\mathbf{f} = \begin{bmatrix} f_0 = 1 \\ f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix}$$

K reprezentaci vzorku  
 $\mathbf{x}$  lze tedy použít jeho  
příznakový vektor...

→ pro trénovací  
vzorek  $(\mathbf{x}^{(i)}, y^{(i)})$ :

$$\mathbf{x}^{(i)} \rightarrow \mathbf{f}^{(i)} = \begin{bmatrix} f_0^{(i)} = 1 \\ f_1^{(i)} = \text{pod}(\mathbf{x}^{(i)}, \mathbf{l}^{(1)}) \\ f_2^{(i)} = \text{pod}(\mathbf{x}^{(i)}, \mathbf{l}^{(2)}) \\ \vdots \\ f_m^{(i)} = \text{pod}(\mathbf{x}^{(i)}, \mathbf{l}^{(m)}) \end{bmatrix}$$

$$f_i^{(i)} = \text{pod}(\mathbf{x}^{(i)}, \mathbf{l}^{(i)}) = \exp\left(-\frac{0}{2\sigma^2}\right) = 1$$



# SVM s jádry

## Výpočet hypotézy a trénování

**Výpočet hypotézy:** Dán vzorek  $\mathbf{x}$ , vypočteme příznakový vektor  $\mathbf{f} \in \mathbb{R}^{m+1}$ . Hypotéza predikuje  $y = 1 \Leftrightarrow \Theta^T \mathbf{f} \geq 0$

$$\underbrace{\Theta_0 f_0 + \Theta_1 f_1 + \cdots + \Theta_m f_m}_{\Theta_0 f_0 + \Theta_1 f_1 + \cdots + \Theta_m f_m \geq 0} \geq 0$$

### Trénování:

$$\min_{\Theta} C \sum_{i=1}^m y^{(i)} \text{Cost}_1(\Theta^T \mathbf{f}^{(i)}) + (1 - y^{(i)}) \text{Cost}_0(\Theta^T \mathbf{f}^{(i)}) + \frac{1}{2} \sum_{j=1}^{n=m} \Theta_j^2$$

ve verzi SVM s jádry používáme vypočítaný příznakový vektor  $\mathbf{f}$  místo původního vektoru  $\mathbf{x}$  z TM

### Matematické triky:

- $\sum_j \Theta_j^2 = \Theta^T \Theta$  (v tom případě se ignoruje  $\Theta_0$ )
- regularizační člen mívá spíše tvar  $\Theta^T M \Theta$  (scaling maticí  $M$ )



# Parametry SVM

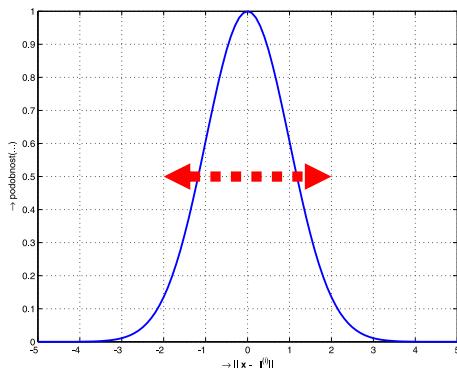
## Vyházení odchylky (*Bias*) a rozptylu (*Variance*)

**C (= 1 / λ)** – **velká hodnota C**: malá odchylka, velký rozptyl  
*(hrozí overfitting)*

– **malá hodnota C**: velká odchylka, malý rozptyl  
*(hrozí underfitting)*

**$\delta^2$  (Gauss)** – **velká hodnota  $\delta^2$** : příznaky  $f_i$  se mění prudčeji,  
menší odchylka, větší rozptyl

– **malá hodnota  $\delta^2$** : příznaky  $f_i$  se mění pomaleji,  
větší odchylka, menší rozptyl





# Použití SVM

## Software/knihovny pro SVM

Numerická optimalizace cenové funkce **SVM** je velmi výpočetně (i algoritmicky) náročná → je rozumné použít již naprogramované vysoce optimalizované (a někdy i akcelerované) knihovny, např. **libsvm** (<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>), **liblinear** (<http://www.csie.ntu.edu.tw/~cjlin/liblinear/>), **Multiclass-SVM** (<http://www.loria.fr/~guermeur/>), ... (ATFG)

Při použití knihovny je obvykle nutno určit:

- hodnotu **regularizačního parametru C**
- jaký **kernel** se užije pro výpočet „podobnosti“ vzorku s landmarksy
  - v závislosti na volbě kernelu je pak nutno ještě specifikovat **parametry příslušného kernelu** (pokud nějaké má)

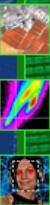




# Použití SVM

## Volba kernelu

- SVM bez kernelu (tzv. **lineární kernel**) –  
 $h_{\Theta}(x) : y = 1 \Leftrightarrow \Theta^T x \geq 0 \dots$  použijeme zejména, je-li  $n$  velké (tj. hodně příznaků) a  $m$  malé (málo trénovacích dat)
  - **gaussovský kernel** –  
měření podobnosti vzorku  $x$  s landmarkem  $I$  prostřednictvím exponenciální funkce (viz předchozí výklad) – v případě tohoto kernelu je třeba vhodně zvolit parametr  $\delta^2$   
... použijeme zejména, je-li  $n$  malé a/nebo  $m$  velké
- SVM s gaussovským kernelem „umí“ vyrobit velmi komplexní tvar rozhodovací hranice, tj. vhodné pro obtížně separovatelná data





# Použití SVM

## Gaussovský kernel

Knihovny někdy vyžadují od uživatele dodat implementaci kernelové funkce, tj. funkce měřící podobnost vzorku  $x$  s landmarkem  $I$ :

```
1 function f = kernel(x1, x2, sigma)
2 -     f = exp(-norm(x1 - x2)^2 / 2 * sigma^2);
3 -     return
```

$f_i^{(i)}$        $x^{(i)}$        $I^{(j)} = x^{(j)}$

### Důležitá poznámka:

Před použitím gaussovského kernelu je nutné provést naškálování příznaků (je to kvůli výpočtu normy, kde mohou být v kvadrátu podstatně nesouměřitelné hodnoty, což drasticky ovlivní výslednou hodnotu)...



# Použití SVM

## Jiné volby kernelů

**Pozor:** Ne všechny funkce kvantifikující podobnost vzorku  $x$  s landmarkem  $I$  podobnost( $x, I$ ) mohou být validním kernelem.

Je nutné, aby funkce vyhovovala tzv. **Mercerově větě**, tzn. musí se jednat o symetrickou pozitivně-definitní funkci – tato podmínka musí být splněna zejména proto, aby pokročilé optimizační algoritmy (využívající některé speciální numerické triky) fungovaly správně a konvergovaly.

### Další použitelné kernely:

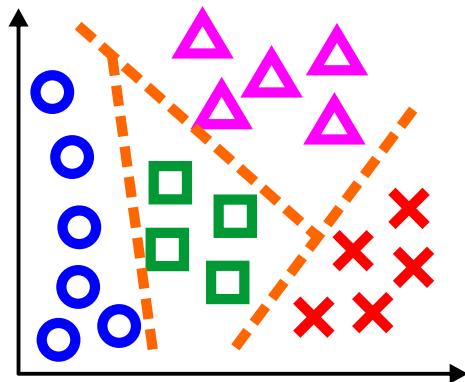
- polynomiální kernel – moc se nepoužívá, je složitý a funguje hůř než gaussovský kernel
- řetězcový kernel (*String Kernel*),  $\chi^2$  (chí-kvadrát) kernel, průnik histogramů, ...



# Použití SVM

## Klasifikace do více tříd

Některé knihovny SVM již disponují schopností klasifikovat do více tříd (např. M-SVM).



Pokud ne, použijeme techniku **one-vs-all**: Natrénujeme  $K$  SVM klasifikátorů, přičemž každý poskytuje hypotézu, zda  $y = i$ , pro  $i = 1, 2, \dots, K$ . Trénováním získáme  $K$  vektorů parametrů  $\Theta^{(1)}$ ,  $\Theta^{(2)}, \dots, \Theta^{(K)}$ . Vybereme třídu  $i$  s nejvyšší hodnotou  $(\Theta^{(i)})^T \mathbf{x}$ .



# Použití SVM

## SVM vs logistická regrese či ANN

$n$  – počet příznaků,  $m$  – počet trénovacích vzorků (velikost TM)

- **$n$  je velké (v porovnání s  $m$ ):**  $n = 10\ 000$ ,  $m = 10 \dots 1\ 000$   
použijeme logistickou regresi nebo SVM bez kernelu
- **$n$  je malé,  $m$  je středně velké:**  $n = 1 \dots 1\ 000$ ,  $m = 10 \dots 10\ 000$   
použijeme SVM s gaussovským kernelem
- **$n$  je malé,  $m$  velké:**  $n = 1 \dots 1\ 000$ ,  $m = 50\ 000+$   
přidáme více příznaků a pak použijeme logistickou regresi  
nebo SVM bez kernelu



Většinu naznačených situací řeší obvykle stejně dobře **neuronová síť (ANN)**, ale tu je mnohem problematičtější a časově náročnější natrénovat.