Support Vector Machines
(úvod)

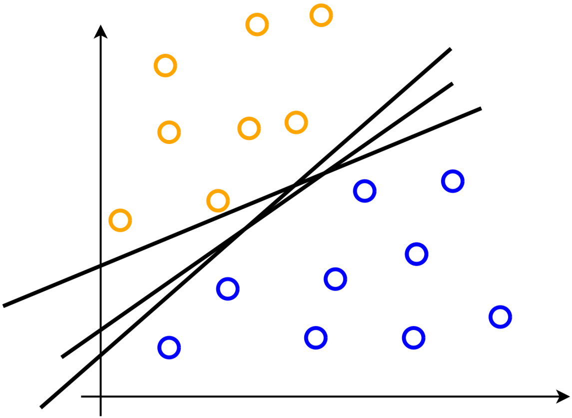
Osnova

- formulace úlohy
- *Support Vector Classifier (SVC)*
  - matematické odvození
- “jaderný trik” (*kernel trick*)
- *Support Vector Machine*
  - *SVM = SVC + kernel*
- hledání parametrů
- rošíření
  - klasifikace s “měkkou hranicí” (*soft-margin classification*)
  - klasifikace do více tříd

Formulace úlohy

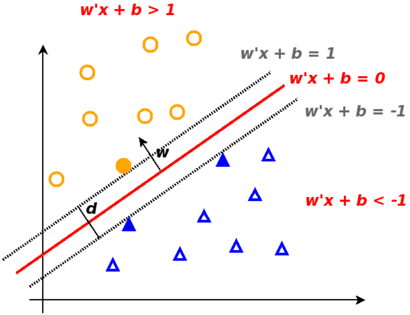
- dáno:
  - $\{\boldsymbol{x_i}, y_i\}_{i=1}^{n}$
  - $\boldsymbol{x_i} \in \mathbb{R}^p$ příznaky
  - $y_i \in \{-1,\ 1\}$ třídy
- úloha: klasifikovat nové $\boldsymbol{x} \in \mathbb{R}^p$ do třídy -1 nebo 1

Motivace



která přímka nejlépe odděluje instance daných dvou tříd?

Přístup SVC (2)



myšlenka:

- maximalizace hranice o šířce $d = \frac{2}{\|\boldsymbol{w}\|}$
- za podmínky správné klasifikace:
  - $\boldsymbol{w}'\boldsymbol{x_i} + b \geq +1$, když $y_i = +1$
  - $\boldsymbol{w}'\boldsymbol{x_i} + b \leq -1$, když $y_i = -1$

Formulace optimalizační úlohy

- $argmax_{\boldsymbol{w},b} \{\frac{2}{\|\boldsymbol{w}\|}\}$
- za podmínky správné klasifikace:
  - $\boldsymbol{w}'\boldsymbol{x_i} + b \geq +1$, když $y_i = +1$
  - $\boldsymbol{w}'\boldsymbol{x_i} + b \leq -1$, když $y_i = -1$
  - nebo ekvivalentně: $y_i(\boldsymbol{w}'\boldsymbol{x_i} + b) \geq 1$

ekvivalentně (formulace 1):

- $argmin_{\boldsymbol{w},b} \{\frac{1}{2}\|\boldsymbol{w}\|^2\}$
- za podmínky správné klasifikace: $y_i(\boldsymbol{w}'\boldsymbol{x_i} + b) \geq 1$

ekvivalentně (chceme se zbavit omezení) (formulace 2):

- $L_p = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{n} \alpha_i[(\boldsymbol{w}'\boldsymbol{x_i} + b)y_i - 1]$
- $L_p = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{n} \alpha_i[(\boldsymbol{w}'\boldsymbol{x_i} + b)y_i] + \sum_{i=1}^{n} \alpha_i$
- $argmin_{\boldsymbol{w},b}\ L_p$
- za podmínky: $\alpha_i \geq 0$
- α_i - Lagrangeovy multiplikátory

Formulace optimalizační úlohy
Primární a duální úloha

$L_p = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{n} \alpha_i[(\boldsymbol{w}'\boldsymbol{x_i} + b)y_i] + \sum_{i=1}^{n} \alpha_i$

argmin$_{\boldsymbol{w},b}$ L_p (za podmínky $\alpha_i \geq 0$)

Požadujeme $\nabla L_p = 0$:

- $\frac{\partial L_p}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{n} \alpha_i \boldsymbol{x_i} y_i = 0 \rightarrow \boldsymbol{w} = \sum_{i=1}^{n} \alpha_i \boldsymbol{x_i} y_i$
- $\frac{\partial L_p}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i = 0$

Substitucí do L_p získáme duální úlohu:

$L_d = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j \boldsymbol{x_i}\boldsymbol{x_j} y_i y_j$

argmax$_{\alpha_i} L_d$ (za podmínky $\alpha_i \geq 0$)

Optimalizační úlohy
Rekapitulace

- formulace 1:
  - $argmin_{\mathbf{w},b} \{\frac{1}{2}\|\mathbf{w}\|^2\}$ (za podm.: $y_i(\mathbf{w}'\mathbf{x_i} + b) \geq 1$)
  - $p + 1$ parametrů, n lin. omezení
- formulace 2:
  - $L_p = -\frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i[(\mathbf{w}'\mathbf{x_i} + b)y_i] + \sum_{i=1}^{n} \alpha_i$
  - $argmin_{\mathbf{w},b}\ L_p$ (za podmínky $\alpha_i \geq 0$)
  - $p + 1$ parametrů, n omezení
- formulace 3:
  - $L_d = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j \mathbf{x_i}\mathbf{x_j} y_i y_j$
  - $argmax_{\alpha_i} L_d$ (za podmínky $\alpha_i \geq 0$)
  - n parametrů, n omezení
  - **proč to všechno snažení?**
  - **data $\mathbf{x_i}$ vystupují pouze ve formě součinů $\mathbf{x_i}\mathbf{x_j}$**
  - většina α_i nulových, $\alpha_i > 0$ právě pro *support vectors*

Řešení optimalizační úlohy

Rozhodovací funkce $f(\boldsymbol{x}) = sign(\boldsymbol{w}'\boldsymbol{x} + b)$
$\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i \boldsymbol{x_i} y_i$
jak získat b?

- pro lib. *support vector*: $y_i(\boldsymbol{w}'\boldsymbol{x_i} + b) = 1$
- $\rightarrow b = \frac{1}{y_i} - \boldsymbol{w}'\boldsymbol{x_i}$
- prakticky: $b = \frac{\sum_{i,\alpha_i>0}(\frac{1}{y_i} - \boldsymbol{w}'\boldsymbol{x_i})}{\sum_{i,\alpha_i>0} 1}$

tedy konečně dostáváme:

- $f(\boldsymbol{x}) = sign(\sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x_i}'\boldsymbol{x} + b)$

Trik s jádrem *(Kernel trick)*

Rozhodovací funkce:

- $f(\boldsymbol{x}) = sign(\sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x_i}' \boldsymbol{x} + b)$

Pozorování:

- příznaky $\boldsymbol{x}$ pouze ve formě skalárním součinu $\boldsymbol{x_i}'\boldsymbol{x}$
- skalární součin $\boldsymbol{r}'\boldsymbol{s}$ lze nahradit jádrem $K(\boldsymbol{r}, \boldsymbol{s})$
- $K(\boldsymbol{r}, \boldsymbol{s}) = \phi(\boldsymbol{r})'\phi(\boldsymbol{s})$
- jádro může realizovat operaci odpovídající skalárnímu součinu ve vysokorozměrném prostoru
- použitím jádra se z SVC stávají SVM

Používaná jádra:

- polynomiální: $K(\boldsymbol{r}, \boldsymbol{s}) = (\boldsymbol{r}'\boldsymbol{s} + 1)^q$
- *Gaussian radial-basis function (RBF)*: $K(\boldsymbol{r}, \boldsymbol{s}) = exp\{-\frac{\|\boldsymbol{r}-\boldsymbol{s}\|^2}{2\sigma^2}\}$
- hyperbolický tangens: $K(\boldsymbol{r}, \boldsymbol{s}) = tanh\{\beta_1 \boldsymbol{r}'\boldsymbol{s} + \beta_2\}$

Trik s jádrem *(Kernel trick)*
Příklad

Polynomiální jádro stupně $q = 2$: $K(\boldsymbol{r}, \boldsymbol{s}) = (\boldsymbol{r}'\boldsymbol{s} + 1)^2$,
$\boldsymbol{r}, \boldsymbol{s} \in \mathbb{R}^2$, operuje v prostoru $\mathbb{R}^6$:
$\boldsymbol{r} \rightarrow \phi(\boldsymbol{r}) = \{r_1^2,\ r_2^2,\ \sqrt{2}r_1 r_2,\ \sqrt{2}r_1,\ \sqrt{2}r_2,\ 1\}$
$\boldsymbol{s} \rightarrow \phi(\boldsymbol{s}) = \{s_1^2,\ s_2^2,\ \sqrt{2}s_1 s_2,\ \sqrt{2}s_1,\ \sqrt{2}s_2,\ 1\}$

$\phi(\boldsymbol{r})'\phi(\boldsymbol{s}) = r_1^2 s_1^2 + r_2^2 s_2^2 + 2r_1 r_2 s_1 s_2 + 2r_1 s_1 + 2r_2 s_2 + 1$

$(\boldsymbol{r}'\boldsymbol{s} + 1)^2 = (r_1 s_1 + r_2 s_2 + 1)^2 =$
$r_1^2 s_1^2 + r_2^2 s_2^2 + 1 + 2r_1 r_2 s_1 s_2 + 2r_1 s_1 + 2r_2 s_2$

Ukázka jader

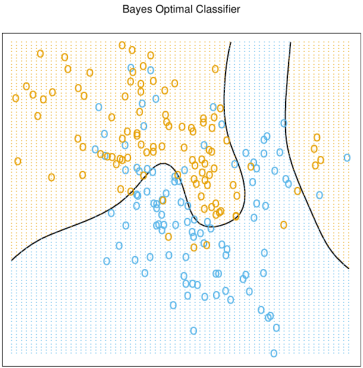
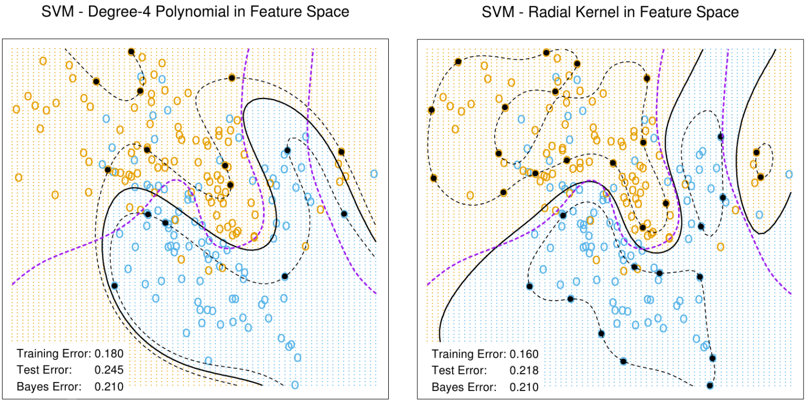


FIGURE 2.5. *The optimal Bayes decision boundary for the simulation example of Figures 2.1, 2.2 and 2.3. Since the generating density is known for each class, this boundary can be calculated exactly (Exercise 2.2).*

(převzato z `http://statweb.stanford.edu/~tibs/ElemStatLearn/`)

Ukázka jader



(převzato z `http://statweb.stanford.edu/~tibs/ElemStatLearn/`)

Klasifikace s “měkkou hranicí”

Soft-margin classification

Motivace:

- třídy nemusejí být oddělitelné
- přesto chceme SVM použít

Řešení:

- dovolit SVM udělat “malou” chybu

Jak:

- $argmin_{\mathbf{w},b} \{\frac{1}{2}\|\mathbf{w}\|^2\} + C\sum_{i=1}^{n} \xi_i$
- za podm. téměř správné klasifikace: $y_i(\mathbf{w}'\mathbf{x_i} + b) \geq 1 - \xi_i$
- C představuje regularizační konstantu
- $C = \infty$ odpovídá původní formulaci separabilní úlohy
- C se hledá nejčastěji pomocí křížové validace

SVM v praxi

- fungují velmi dobře
- časová složitost trénování: $O(n^2)$
- uživatel volí typ jádra a parametry
- parametry se hledají typicky křížovou validací
- po natrénování si stačí pamatovat *support vectors*

Porovnání metod

TABLE 10.1. *Some characteristics of different learning methods. Key:* ▲*= good,* ◆*=fair, and* ▼*=poor.*

Characteristic	Neural Nets	SVM	Trees	MARS	k-NN, Kernels
Natural handling of data of "mixed" type	▼	▼	▲	▲	▼
Handling of missing values	▼	▼	▲	▲	▲
Robustness to outliers in input space	▼	▼	▲	▼	▲
Insensitive to monotone transformations of inputs	▼	▼	▲	▼	▼
Computational scalability (large N)	▼	▼	▲	▲	▼
Ability to deal with irrelevant inputs	▼	▼	▲	▲	▼
Ability to extract linear combinations of features	▲	▲	▼	▼	◆
Interpretability	▼	▼	◆	▲	▼
Predictive power	▲	▲	▼	◆	▲

(převzato z `http://statweb.stanford.edu/~tibs/ElemStatLearn/`)

Klasifikace do více tříd

SVM umí rozlišovat jen do dvou tříd

možná řešení klasifikace do K tříd:

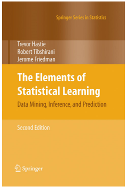
- "jeden proti všem": K úloh: klasifikace třídy k proti zbytku, "vítěz bere vše"
- "jeden na jednoho": $\frac{K(K-1)}{2}$ úloh: klasifikace třídy k_1 proti k_2, hlasování

Rozšíření SVM

- *SVM regression*
- shlukování založené na SVM
- detekce nečekaných pozorování *(novelty detection)*

Literatura

Christopher J.C. Burges: A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery (1998), volume 2, p.121-167.



Hastie T., Tibshirani R., Friedman J.: **The Elements of Statistical Learning** (2009). Springer-Verlag, `http://statweb.stanford.edu/~tibs/ElemStatLearn/`