EVROPSKÁ UNIE
esf
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
OP Vzdělávání pro konkurenceschopnost
2007-13
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Strukturní kasifikátory a jejich aplikace v počítačovém vidění

Ing. Vojtěch Franc, Ph.D. (ČVUT v Praze)

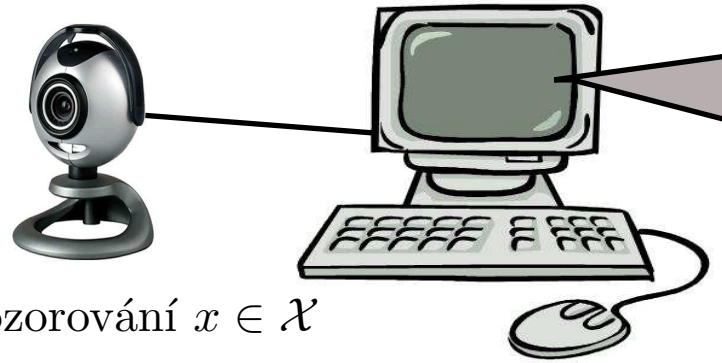
17. prosince 2012

Tato prezentace je spolufinancována Evropským sociálním fondem a státním rozpočtem České republiky.

Rozpoznávání, učení, optimalizace

Objekt

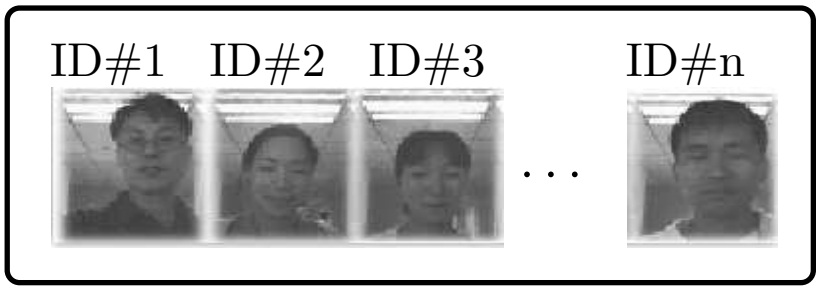
Klasifikátor $y = h(x)$



Pozorování $x \in \mathcal{X}$

User ID #1
User ID #2
...

Skrytý stav $y \in \mathcal{Y}$



Trénovací příklady

Učení

$y = h(x; \theta^*)$

Klasifikátor

Optimalizace

$\theta^* = \text{argmin}_{\theta \in \Theta} F(\theta)$

Co to je strukturní klasifikace?

- Metody strukturního rozpoznávání se používají v případech, kdy je vhodné modelovat zkoumaný objekt jako množinu jednodušších objektů a vztahů mezi nimi, kde tyto vztahy tvoří nějakou strukturu.

- Zkoumaný objekt je popsán:
  1. Sekvencí pozorování $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathcal{X}$.
  2. Sekvencí skrytých stavů $\boldsymbol{y} = (y_1, \ldots, y_m) \in \mathcal{Y}$.

- Nechť $\mathcal{I} = \{1, \ldots, n\}$ a $\mathcal{J} = \{1, \ldots, m\}$ jsou množiny indexů jednotlivých pozorování, resp. skrytých stavů.

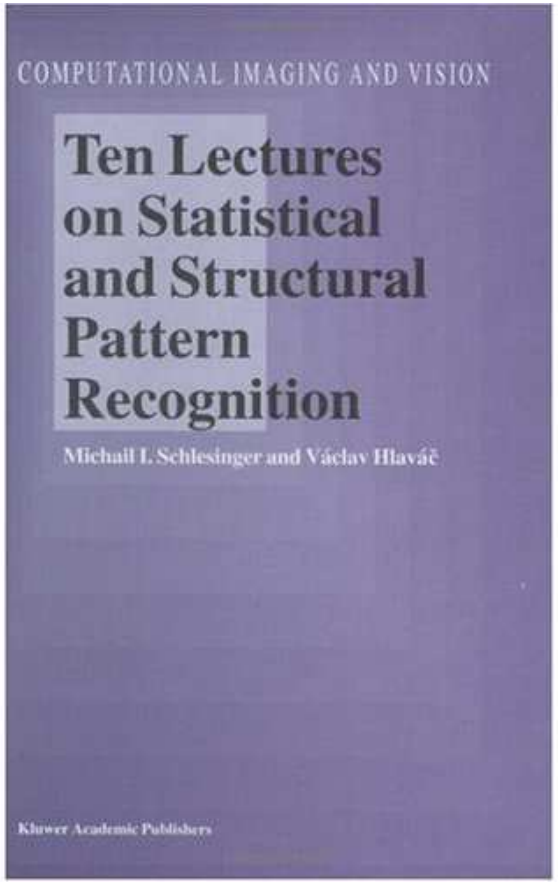
- Doménou strukturního rozpoznávání jsou aplikace, kdy množiny $\mathcal{I}$ a $\mathcal{J}$ mají jasnou strukturu, např. tvoří řetězec nebo dvourozměrnou mřížku.

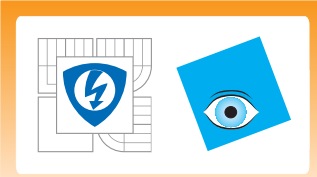
Plán kurzu

1. **Aplikace**
   (a) Řešení hlavolamu Sudoku.
   (b) Max-sum klasifikátor.
   (c) Detekce významných bodů na lidské tváři.
   (d) Segmentace a rozpoznávání textu.
   (e) Segmentace obrazu.
2. **Algoritmy učení**
   (a) Formulace baysovské klasifikační úlohy.
   (b) Generativní a diskriminativní metody učení.
   (c) Support Vector Machines.
   (d) Structured Output Support Vector Machines.
3. **Optimalizace**
   (a) Zobecnění gradientu pro nehladké funkce.
   (b) Zobecněná gradientní metoda.
   (c) Metody odsekávajících nadrovin.

Učebnice struktuního rozpoznávání

Schlesinger, Hlaváč. Deset přednášek z teorie statistického a strukturního rozpoznávání. Vydavatelství ČVUT, Praha 1999.





Část I: Aplikace

Plán přednášky

1. Aplikace I: Řešení hlavolamu Sudoku
2. Max-sum klasifikátor
3. Aplikace II: Detekce významných bodů na lidské tváři
4. Aplikace III: Segmentace a rozpoznávání textu
5. Aplikace IV: Segmentace obrazu

Aplikace I: Řešení hlavolamu Sudoku

Sudoku

Zadání hlavolamu

					8			
	1	9	5	6		2		
2	5			1		3	6	
9					2		8	1
	8	2	6		9			
5	7		1					2
	2	1		9			4	3
		5		7	6	8		
8	9		3					

Řešení hlavolamu

7	6	3	4	2	8	1	9	5
4	1	9	5	6	3	2	7	8
2	5	8	9	1	7	3	6	4
9	3	4	7	5	2	6	8	1
1	8	2	6	3	9	4	5	7
5	7	6	1	8	4	9	3	2
6	2	1	8	9	5	7	4	3
3	4	5	2	7	6	8	1	9
8	9	7	3	4	1	5	2	6

- Cílem hry je doplnit hrací pole tak aby v každém sloupci, v každém řádku a každém poli 3×3 byla čísla $\{1, \ldots, 9\}$.

Řešení Sudoku pomocí strukturní klasifikace

- Na řešení hlavolamu budeme pohlížet jako na klasifikační úlohu:

$$\text{řešení hlavolamu} = h\Big(\text{ zadání hlavolamu}\Big)$$

- Model Sudoku hry:
  - $\mathcal{T} = \{(i,j) \in \mathcal{N}^2 \mid 1 \leq i \leq 9, 1 \leq j \leq 9\}$ hrací pole.
  - Vstup hry je $\boldsymbol{x} = (x_t \in \{\square, 1, \ldots, 9\} \mid t \in \mathcal{T}) \in \mathcal{X}$.
  - Výstup je $\boldsymbol{y} = (y_t \in \{1, \ldots, 9\} \mid t \in \mathcal{T}) \in \mathcal{Y}$ splňující pravidla hry.
- Chceme naleznout rozhodovací funkci, strukturní klasifikátor,

$$h\colon \mathcal{X} \to \mathcal{Y}\,,$$

  která vyřeší každý zadaný hlavolam.

Jak by měl Sudoku klasifikátor vypadat?

- Typická reprezentace klasifikátoru $f : \mathcal{X} \to \mathcal{Y}$:

$$h(\boldsymbol{x}) = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}} f(\boldsymbol{x}, \boldsymbol{y})$$

  kde $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ je **skórovací funkce měřící kvalitu páru** $(\boldsymbol{x}, \boldsymbol{y})$.

- Nechť $\mathcal{T}_i \subset \mathcal{T}$, $i = 1, \ldots, 27$ jsou podmnožiny polí, která musí obsahovat čísla $\{1, \ldots, 9\}$, tj. všechny řádky, sloupce a pole 3×3.
- Skórovací funkce Sudoku klasifikátoru by měla splňovat tyto podmínky:
  1. $f(\boldsymbol{x}, \boldsymbol{y}) = -\infty$ pokud existuje $t \in \mathcal{T}$ takové, že $x_t \neq \square \wedge x_t \neq y_t$.
  2. $f(\boldsymbol{x}, \boldsymbol{y}) = -\infty$ pokud existuje $i \in \{1, \ldots, 27\}$ takové, že $\{y_t \mid t \in \mathcal{T}_i\} \neq \{1, \ldots, 9\}$
  3. $f(\boldsymbol{x}, \boldsymbol{y}) = 0$ ve všech ostatních případech.

Řešení Sudoku pomocí strukturní klasifikace

- Zaveďme množinu sousedících (vzájemně se ovlivňujících) objektů

$$\mathcal{E} = \{\{(i,j),(i',j')\} \mid i = i' \vee j = j' \vee (\lceil i/3 \rceil = \lceil i'/3 \rceil \wedge \lceil j/3 \rceil = \lceil j'/3 \rceil)\}$$

- Zaveďme funkci $g\colon \{1,\ldots,9\}^2 \to \{0,-\infty\}$,

$$g(y,y') = \begin{cases} 0 & \text{pokud} \quad y \neq y' \\ -\infty & \text{pokud} \quad y = y' \end{cases}$$

a funkci $q\colon \{\square,1,\ldots,9\} \times \{1,\ldots,9\} \to \{0,-\infty\}$

$$q(x,y) = \begin{cases} 0 & \text{pokud} \quad x = \square \\ -\infty & \text{pokud} \quad x \neq \square \wedge y \neq x \end{cases}$$

Řešení Sudoku pomocí strukturní klasifikace

- Skórovací funkci Sudoku klasifikátoru můžeme reprezentovat jako součet funkcí arity 2:

$$f(\boldsymbol{x}, \boldsymbol{y}) = \underbrace{\sum_{t \in \mathcal{T}} q(x_t, y_t)}_{\text{vyplněná políčka se kopírují}} + \underbrace{\sum_{\{t,t'\} \in \mathcal{E}} g(y_t, y_{t'})}_{\text{řešení splňuje pravidla}}$$

- Řešení Sudoku hlavolamu se zadáním $\boldsymbol{x}$ získáme ze strukturního klasifikátoru

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}} f(\boldsymbol{x}, \boldsymbol{y})$$

- *Poznámka:* Ohodnocení klasifikátoru vyžaduje řešit tzv. max-sum problém, který je obecně NP-těžký.

Učení Sudoku strukturního klasifikátoru

- **Cíl učení:** nalézt Sudoku klasifikátor (tj. pochopit pravidla hry), na základě příkladů, které získáme z pozorování hry zkušeného Sudoku hráče?
- Přepodkládejme, že je k dispozici **množina trénovacích příkladů**:

$$\mathcal{D} = \{(\boldsymbol{x}^1, \boldsymbol{y}^1), \ldots, (\boldsymbol{x}^m, \boldsymbol{y}^m))\} \in (\mathcal{X} \times \mathcal{Y})^m$$

  tj. množina obsahuje dvojice (zadání hlavolamu, řešení hlavolamu).

- Současné algoritmy učení zvládají následující úlohu:
  - Struktura klasifikátoru je známá, tj. množina objektů $\mathcal{T}$ a struktura sousedství $\mathcal{E}$ je daná.
  - Neznámé funkce q a g naučíme z množiny příkladů $\mathcal{D}$.

Učení Sudoku strukturního klasifikátoru

- Funkce q a g budeme reprezentovat vektorem $\boldsymbol{\theta} \in \mathbb{R}^n$, který má $n = 90 + 81$ složek, tj. složky vektoru $\boldsymbol{\theta}$ jsou čísla $q(x, y)$ a $g(y, y')$, $x \in \mathcal{X}$, $y \in \mathcal{Y}$, $y' \in \mathcal{Y}$.

- V tomto případě je Sudoku klasifikátor instancí **lineárního klasifikátoru**:

$$\begin{aligned} h(\boldsymbol{x}; \boldsymbol{\theta}) &= \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}} \left[\sum_{t \in \mathcal{T}} q(x_t, y_t) + \sum_{\{t,t'\} \in \mathcal{E}} g(y_t, y_{t'}) \right] \\ &= \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}} \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}) \rangle \end{aligned}$$

kde $\boldsymbol{\Psi} \colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^n$ je funkce složená s indikátorových funkcí.

Učení Sudoku strukturního klasifikátoru

- **Úloha učení:** nalezni parametry $\boldsymbol{\theta} \in \mathbb{R}^n$, tak aby klasifikátor $h(\boldsymbol{x}; \boldsymbol{\theta})$ udělal na trénovací množině $\mathcal{D} = \{(\boldsymbol{x}^1, \boldsymbol{y}^1), \ldots, (\boldsymbol{x}^m, \boldsymbol{y}^m))\}$ co nejméně chyb.
- Zaveďme ztrátovou funkci $\Delta \colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, tak že

$$\Delta(\boldsymbol{y}, \boldsymbol{y}') = \begin{cases} 0 & \text{pokud} \quad \boldsymbol{y} = \boldsymbol{y}' \\ 1 & \text{pokud} \quad \boldsymbol{y} \neq \boldsymbol{y}' \end{cases}$$

- Učení Sudoku klasifikátoru formulujeme jako optimalizační problém

$$\boldsymbol{\theta}^* \in \operatorname*{Argmin}_{\boldsymbol{\theta} \in \mathbb{R}^n} \underbrace{\sum_{i=1}^{m} \Delta(\boldsymbol{y}^i, h(\boldsymbol{x}^i; \boldsymbol{\theta}))}_{\text{počet chyb na trénovací množině}}$$

Učení Sudoku strukturního klasifikátoru: numerické výsledky, učení z 18 příkladů

$q(y, x) = q(y)$ pro $y = x$ nebo $x = \square$ a $q(x, y) = q_{\neq}$ pro $x \neq y$ a $x \neq \square$

$q_{\neq}$	y	1	2	3	4	5	6	7	8	9
-1870	$q(y)$	867	295	696	234	224	339	228	455	378

$g(y, y')$

y/y'	1	2	3	4	5	6	7	8	9
1	-552	19	47	66	65	73	61	66	73
2	84	-500	67	57	71	56	79	60	65
3	78	30	-437	48	48	47	68	50	54
4	73	-28	63	-266	55	53	76	56	60
5	23	67	57	47	-397	46	68	49	52
6	74	66	58	49	49	-448	69	49	54
7	-26	9	28	49	48	47	-342	48	54
8	36	77	60	49	49	46	67	-441	52
9	79	20	57	48	46	48	65	51	-441

Max-sum klasifikátor

Max-sum klasifikátor

$(\mathcal{T}, \mathcal{E})$	...	neorientovaný graf popisující strukturu objektu
$\mathcal{T}$	...	je množina objektů
$\mathcal{E} \subseteq \binom{\mathcal{T}}{2}$	...	je množina sousedů (sousedících objektů)
$\boldsymbol{x} = (x_t \in X \mid t \in \mathcal{T})$	...	je $\lvert\mathcal{T}\rvert$-tice pozorování, tj., $\mathcal{X} = X^{\mathcal{T}}$
$\boldsymbol{y} = (y_t \in Y \mid t \in \mathcal{T})$	...	je $\lvert\mathcal{T}\rvert$-tice skrytých stavů, tj., $\mathcal{Y} = Y^{\mathcal{T}}$
$q_t \colon X \times Y \times \mathbb{R}^n \to \mathbb{R}$	...	kvalita páru (x_t, y_t) pokud je parametr $\boldsymbol{\theta}$
$q_{tt'} \colon Y \times Y \times \mathbb{R}^n \to \mathbb{R}$	...	kvalita páru $(y_t, y_{t'})$ pokud je parametr $\boldsymbol{\theta}$

Max-sum klasifikátor $h \colon \mathcal{X} \to \mathcal{Y}$

$$h(\boldsymbol{x}; \boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}} f(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}) := \left(\sum_{t \in \mathcal{T}} q_t(x_t, y_t; \boldsymbol{\theta}) + \sum_{tt' \in \mathcal{E}} q_{tt'}(y_t, y_{t'}; \boldsymbol{\theta}) \right)$$

Statistická interpretace Max-sum klasifikátoru

- Nechť jsou pozorování $\boldsymbol{x} \in \mathcal{X}$ a skryté stavy $\boldsymbol{y} \in \mathcal{Y}$ realizací náhodných veličin

$$\tilde{\boldsymbol{X}} = (\tilde{X}_t \mid t \in \mathcal{T}), \qquad \tilde{\boldsymbol{Y}} = (\tilde{Y}_t \mid t \in \mathcal{T})$$

  se sdruženou hustota pravděpodobnosti $\tilde{\boldsymbol{X}}$ a $\tilde{\boldsymbol{Y}}$ (Gibbsovo rozdělení)

$$p(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp f(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$$

- Maximálně A Posteriorní (MAP) odhad skrytého stavu

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}} p(\boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{\theta})$$

  dává stejný odhad jako max-sum klasifikátor se skórovací funkcí $f(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$.

Skryté náhodné markovské pole

- Sdružená hustota pravděpodobnosti náhodných proměnných $\tilde{\boldsymbol{X}}$ a $\tilde{\boldsymbol{Y}}$ je obecně funkce $p\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ určená $(|\mathcal{X}| \cdot |\mathcal{Y}| - 1)$ čísly.
- Nechť $\mathcal{N}_t = \{t' \in \mathcal{T} \mid \{t, t'\} \in \mathcal{E}\}$ je množina sousedů objektu t pro daný neorientovaný graf $(\mathcal{T}, \mathcal{E})$.
- Náhodný proces $\tilde{\boldsymbol{X}}$ a $\tilde{\boldsymbol{Y}}$ je skryté markovské pole pokud:
  - $p(x_t \mid \{x_{t'} \mid t' \in \mathcal{T} \setminus \{t\}\}, \boldsymbol{y}) = p(x_t \mid y_t)$
  - $p(y_t \mid \{y_{t'} \mid t' \in \mathcal{T} \setminus \{t\}\}, \boldsymbol{x}) = p(y_t \mid \{y_{t'} \mid t \in \mathcal{N}_t\}, x_t)$.
- Pokud má každý úplný podgraf grafu $(\mathcal{T}, \mathcal{E})$ maximálně dva uzly, pak je sdružená h.p. náhodných proměnných tvořících skryté markovské pole

$$p(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left(\sum_{t \in \mathcal{T}} q_t(x_t, y_t; \boldsymbol{\theta}) + \sum_{tt' \in \mathcal{E}} q_{tt'}(y_t, y_{t'}; \boldsymbol{\theta})\right)$$

Ohodnocení max-sum klasifikátoru vyžaduje řešit max-sum problém

- Výpočet odezvy max-sum klasifikátoru je obecně NP-těžká úloha celočíselného programování:

$$h(\boldsymbol{x};\boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{y}\in\mathcal{Y}} \left[\sum_{t\in\mathcal{T}} q_t(x_t, y_t) + \sum_{\{t,t'\}\in\mathcal{E}} g_{tt'}(y_t, y_{t'}) \right]$$

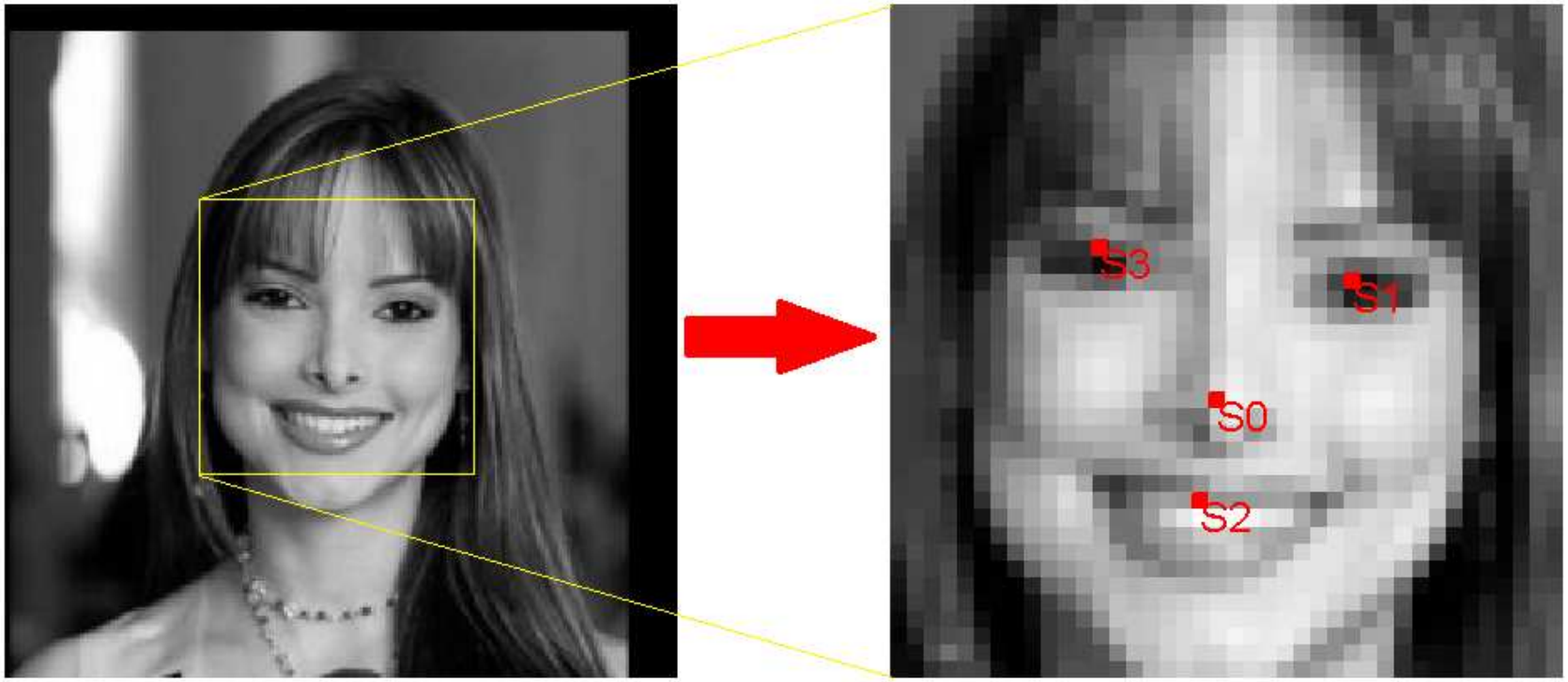
Existují podtřídy max-sum problému, které lze řešit efektivně:

1. Množina sousedů $\mathcal{E}$ tvoří acyklický graf nebo tzv. simple-net.
   - Řeší se dynamickým programováním.
2. Funkce $g_{tt'}$ jsou supermodulární.
   - Y je plně uspořádaná. Lze převést na hledání maximálního toku v síti.
3. Problémy s triviálním ekvivalentem.
   - Řešení lze nalézt pomocí Schlesingerovy LP relaxace.

Aplikace II: Detekce významných bodů na lidské tváři

Detekce významných bodů na tváři

- Cílem je lokalizovat definovanou množinu bodů na lidské tváři jako jsou oči, nos či ústa.

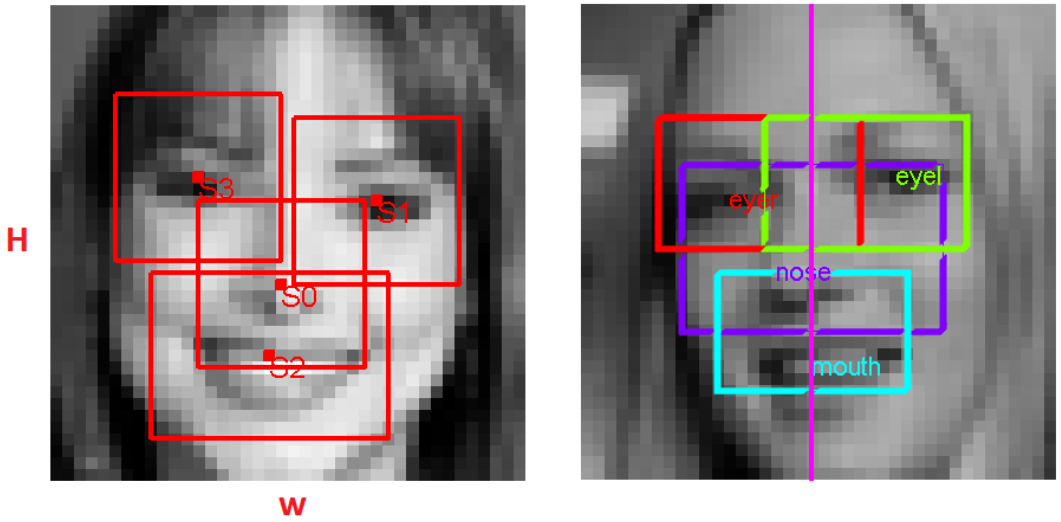


Detekce významných bodů na tváři

- $\boldsymbol{x} \in \mathcal{X}$ je šedotónový obrázek velikosti $[H \times W]$
- $\mathcal{T} = \{0, \ldots, L-1\}$ množina L významných bodů
- $S_t \subset \{1, \ldots, H\} \times \{1, \ldots, W\}$ množina přípustných pozic bodu t
- $\boldsymbol{y} = (\boldsymbol{s}_0, \ldots, \boldsymbol{s}_{L-1}) \in \mathcal{Y} = S_0 \times \cdots \times S_{L-1}$ jsou pozice všech L významných bodů v obrázků

- Zaveďme funkci $q_t(\boldsymbol{x}, \boldsymbol{s}_t)$ jejíž hodnota je úměrná "pravděpodnosti", že významný bod $t \in \mathcal{T}$ je v obrázku $\boldsymbol{x}$ na souřadnici $\boldsymbol{s}_t \in S_t$.
  - Funkce $q_t(\boldsymbol{x}, \boldsymbol{s}_t)$ může např. porovnávat "template" významného bodu s obrázkem.

Detekce významných bodů na tváři



- Pozice významných bodů $\boldsymbol{y} = \{\hat{\boldsymbol{s}}_0, \ldots, \hat{\boldsymbol{s}}_{M-1}\}$ v obrázku $\boldsymbol{x}$ lze nálézt pomocí L nézávislých detektorů

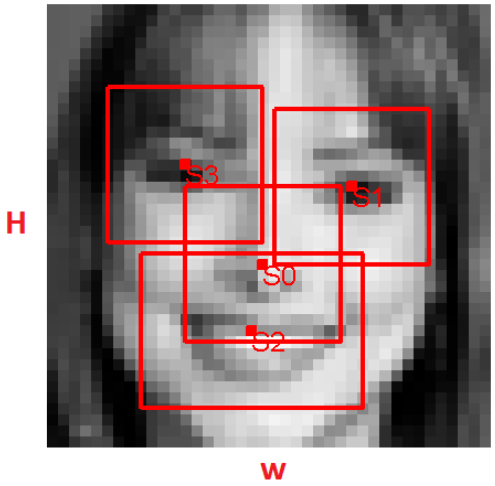
$$\hat{\boldsymbol{s}}_t = \underset{\boldsymbol{s} \in S_t}{\operatorname{argmax}}\, q_t(\boldsymbol{x}, \boldsymbol{s}) \qquad t \in \mathcal{T}$$

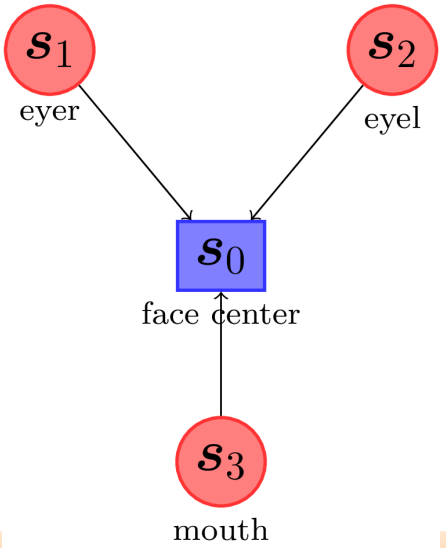
- **Nevýhoda:** nefunguje na "těžkých" obrázcích, protože nevyužívá závislost pozice jednotlivých významných bodů.

Detekce významných bodů na tváři pomocí “Deformable part model”

- $G = (\mathcal{T}, \mathcal{E})$ graf definující významné body a jejich sousedství
- $g_{tt'}(\boldsymbol{s}_t, \boldsymbol{s}_{t'})$ kvalita konfigurace $(\boldsymbol{s}_t, \boldsymbol{s}_{t'})$ sousedících bodů $\{t, t'\} \in \mathcal{E}$
- $f(\boldsymbol{x}, \boldsymbol{y})$ kvalita konfigurace všech významných bodů $\boldsymbol{y}$ v obrázku $\boldsymbol{x}$

$$f(\boldsymbol{x}, \boldsymbol{y}) = \underbrace{\sum_{t \in \mathcal{T}} q_t(\boldsymbol{x}, \boldsymbol{s}_t)}_{\text{shoda s obrázkem}} + \underbrace{\sum_{\{t,t'\} \in \mathcal{E}} g_{tt'}(\boldsymbol{s}_t, \boldsymbol{s}_{t'})}_{\text{apriorní znalost}}$$





Detekce významných bodů na tváři

- Odhad nejlepší konfigurace významných bodů vede na max-sum problém

$$\begin{aligned}\hat{\boldsymbol{y}} &= \operatorname*{argmax}_{\boldsymbol{y}\in\mathcal{Y}} f(\boldsymbol{x},\boldsymbol{y}) \\ &= \operatorname*{argmax}_{(\boldsymbol{s}_0,\ldots,\boldsymbol{s}_{L-1})\in S_0\times\cdots\times S_{L-1}} \left[\sum_{t\in\mathcal{T}} q_t(\boldsymbol{x},\boldsymbol{s}_t) + \sum_{\{t,t'\}\in\mathcal{E}} g_{tt'}(\boldsymbol{s}_t,\boldsymbol{s}_{t'})\right]\end{aligned}$$

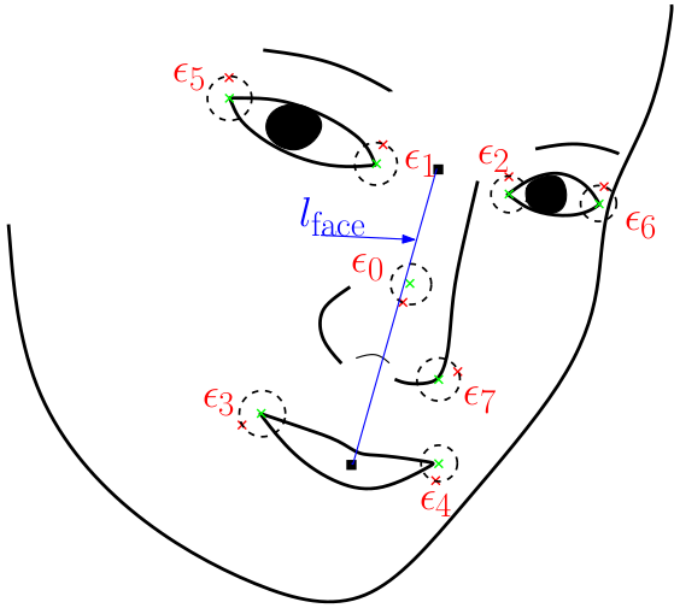
- Pro acyklický graf $(\mathcal{T},\mathcal{E})$ lze nalézt maximální konfiguraci pomocí dynamického programování.

- Jak určit funkce q_t a $g_{tt'}$?

Učení detektoru významných bodů na tváři

- Funkce q_t a $g_{tt'}$ lze učit z trénovací množiny $\mathcal{D} = \{(\boldsymbol{x}^1, \boldsymbol{y}^1), \ldots, (\boldsymbol{x}^m, \boldsymbol{y}^m)\}$, která obsahuje příklady obrázků $\boldsymbol{x}^i$ a jejich manuální anotaci $\boldsymbol{y}^i = (\boldsymbol{s}^i_0, \ldots, \boldsymbol{s}^i_{L-1})$.
- **Formulace učení:** nalezni takové funkce q_t a $g_{tt'}$, aby průměrná odchylka odhadu pozice byla co nejmenší.
- Nechť je odchylka odhadu pozic významných bodů definovaná jako

$$\text{odchylka} = \frac{\varepsilon_0 + \cdots + \varepsilon_{L-1}}{L} \cdot \frac{1}{l_{face}}$$



Učení detektoru významných bodů na tváři

- Definujme ztrátovou funkci $\Delta\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, tak že

$$\Delta(\boldsymbol{y}, \boldsymbol{y}') = \frac{1}{L \cdot l_{\text{face}}(\boldsymbol{y})} \sum_{t=0}^{L-1} \|\boldsymbol{s}_t - \boldsymbol{s}'_t\|$$

  kde $\boldsymbol{y} = (\boldsymbol{s}_0, \ldots, \boldsymbol{s}_{L-1})$ je manuální anotace a $\boldsymbol{y}' = (\boldsymbol{s}_0, \ldots, \boldsymbol{s}_{L-1})$ jsou odhadnuté pozice.

- Funkce q_t a $g_{tt'}$ budeme parametrizovat jako **lineární funkce**

$$q_t(\boldsymbol{x}, \boldsymbol{s}_t) = \langle \boldsymbol{\theta}, \boldsymbol{\Psi}^q(\boldsymbol{x}, \boldsymbol{s}_t) \rangle \qquad g_{tt'}(\boldsymbol{s}_t, \boldsymbol{s}_{t'}) = \langle \boldsymbol{\theta}, \boldsymbol{\Psi}^g(\boldsymbol{s}_t, \boldsymbol{s}_{t'}) \rangle$$

  kde $\boldsymbol{\theta} \in \mathbb{R}^n$ jsou parametry , $\boldsymbol{\Psi}^q(\boldsymbol{x}, \boldsymbol{s}_t)$ jsou příznaky kolem bodu $\boldsymbol{s}_t$ a

$$\boldsymbol{\Psi}^g(\boldsymbol{s}_t, \boldsymbol{s}_{t'}) = (dx, dy, dx^2, dy^2) \quad \text{kde} \quad (dx, dy) = (x_t, y_t) - (x_{t'}, y_{t'})$$

Učení detektoru významných bodů na tváři

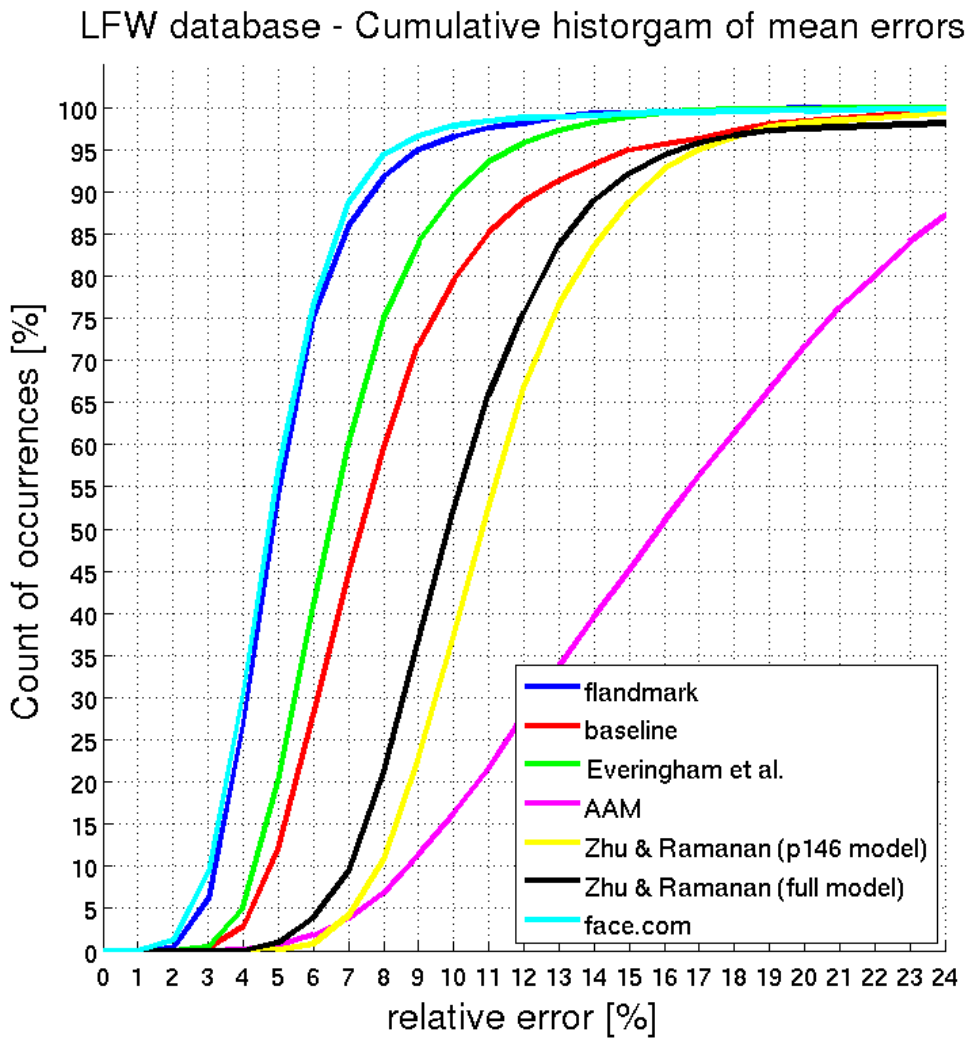
- Při lineární parametrizaci funkcí q_t a $g_{tt'}$ je **max-sum klasifikátor instancí lineárního klasifikátoru**

$$\begin{aligned} h(\boldsymbol{x};\boldsymbol{\theta}) &= \operatorname*{argmax}_{\boldsymbol{y}\in\mathcal{Y}} \left[\sum_{t\in\mathcal{T}} \langle \boldsymbol{\theta}, \boldsymbol{\Psi}^q(\boldsymbol{x}, \boldsymbol{s}_t) \rangle + \sum_{\{t,t'\}\in\mathcal{E}} \langle \boldsymbol{\theta}, \boldsymbol{\Psi}^g(\boldsymbol{s}_t, \boldsymbol{s}_{t'}) \rangle \right] \\ &= \operatorname*{argmax}_{\boldsymbol{y}\in\mathcal{Y}} \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}) \rangle \end{aligned}$$

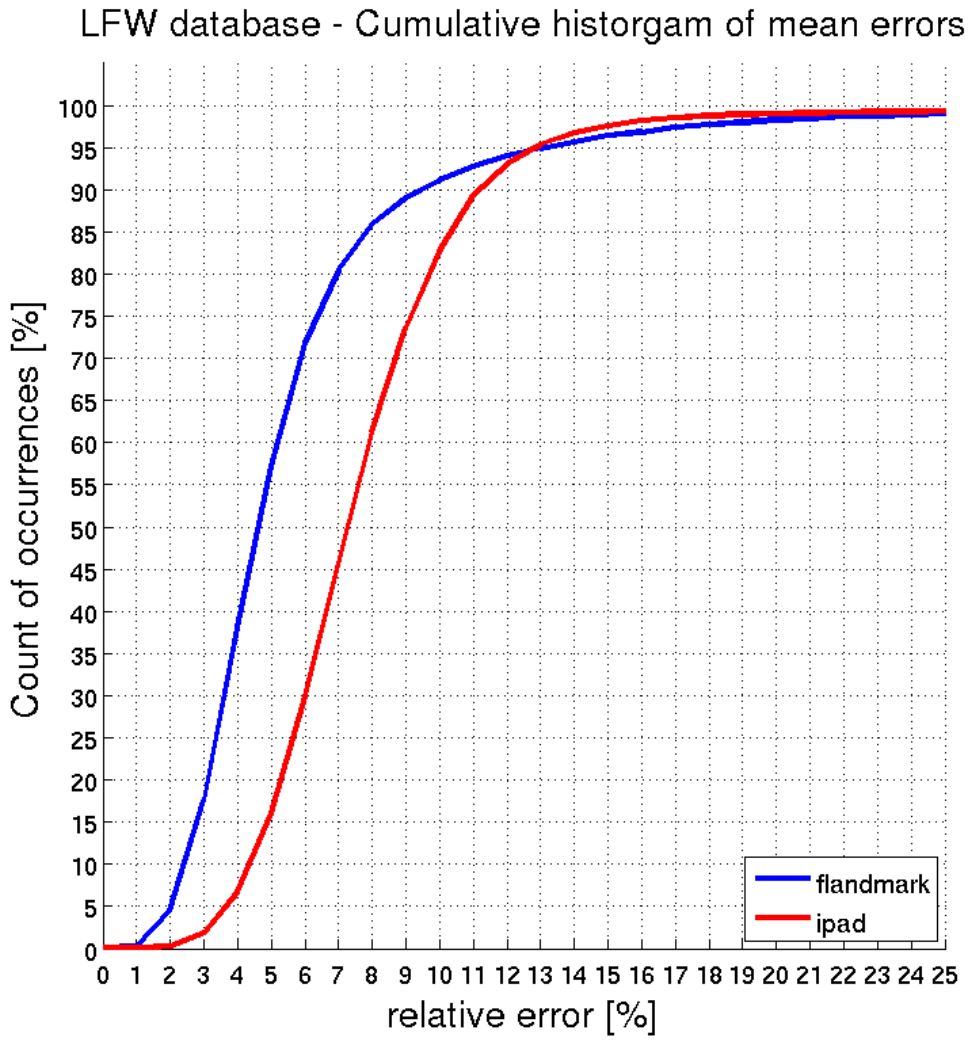
- **Učení klasifikátoru** můžeme formulovat jako **optimalizační problém**

$$\boldsymbol{\theta} = \operatorname*{argmin}_{\boldsymbol{\theta}\in\Theta} \frac{1}{m} \sum_{i=1}^{m} \Delta(\boldsymbol{y}^i, h(\boldsymbol{x}^i;\boldsymbol{\theta}))$$

Porovnání s existujícími detektory



Porovnání s IPAD SDK detektorem



Detekce významných bodů na tváři

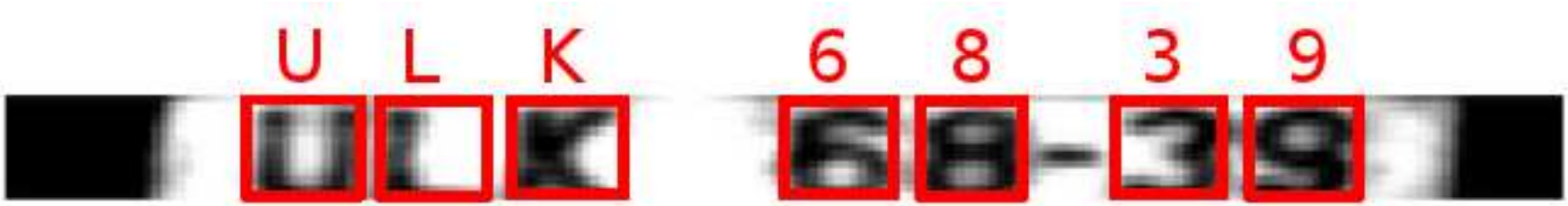
- FLANDMAK
  `http://cmp.felk.cvut.cz/~uricamic/flandmark/`
- Demo detektoru + aplikace v rozpoznávání tváří.

Aplikace III: Segmentace a rozpoznávání textu

Segmentace a rozpoznávání textu

- Pro zadaný obrázek obsahující řádek textu

chceme nalézt pozici znaků a rozpoznat je:



- Klasický přístup sestává ze dvou kroků:
  1. Segmentace obrázku na oblasti znaků.
  2. Rozpoznání jednotlivých znaků.

  Problém: pro správnou segmentaci musíme vědět co hledáme.
- Šlo by segmentovat a rozpoznávat znaky současně jedním klasifikátorem ?

Segmentace a rozpoznávání textu

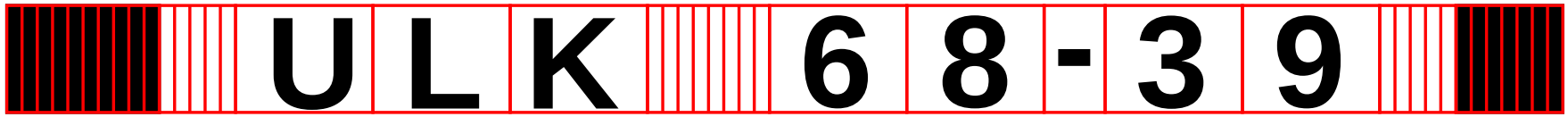
- Obrázek $\boldsymbol{x} \in \mathcal{X}$ velikosti $[W \times H]$ modelujeme jako sekvenci obrázků ("templates") vybraných z konečné množiny $\boldsymbol{\theta} = \{\boldsymbol{\theta}_a \in \mathbb{R}^{H \times \omega(a)} | a \in \mathcal{A}\}$

- Nechť $\boldsymbol{y} = (s_1, \ldots, s_L)$ je segmentací obrázku, kde každý segment $s = (a, k)$, popisuje jméno znaku $a \in \mathcal{A}$ a jeho pozici $k \in \{1, \ldots, W\}$.
- Přípustná segmentace $\boldsymbol{y} \in \mathcal{Y}$ je taková, která pokryje celý obrázek nepřekrývajícími se segmenty, tj. každá $\boldsymbol{y} = (s_1, \ldots, s_L) \in \mathcal{Y}$ splňuje

$$\begin{aligned} k(s_1) &= 1 \\ W &= k(s_L) + \omega(s_L) - 1 \\ k(s_i) &= k(s_{i-1}) + \omega(s_{i-1}) \,, \; i = 2, \ldots, L \end{aligned}$$

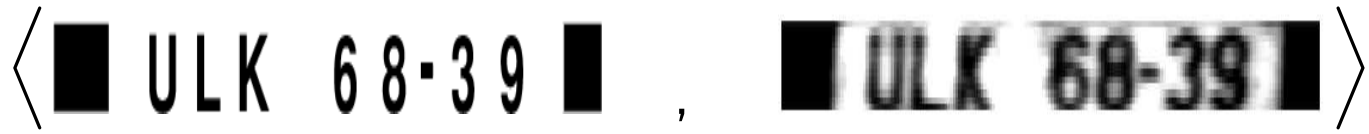
Segmentace a rozpoznávání textu

- Jsou-li dány vzory $\boldsymbol{\theta}$, můžeme pro každou segmentaci $\boldsymbol{y} \in \mathcal{Y}$ vygenerovat syntetický obrázek:



- Podobnost mezi vstupním obrázkem $\boldsymbol{x}$ a synteticky vygenerovaným obrázkem z $\boldsymbol{\theta}$ a segmentace $\boldsymbol{y}$ můžeme měřit jejich korelací:

$$f(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}) = \underbrace{\sum_{i=1}^{L(y)} \sum_{j=1}^{\omega(a(s_i))} \langle \mathrm{col}(\boldsymbol{x}, j + k(s_i) - 1), \mathrm{col}(\boldsymbol{\theta}_{a(s_i)}, j) \rangle}$$

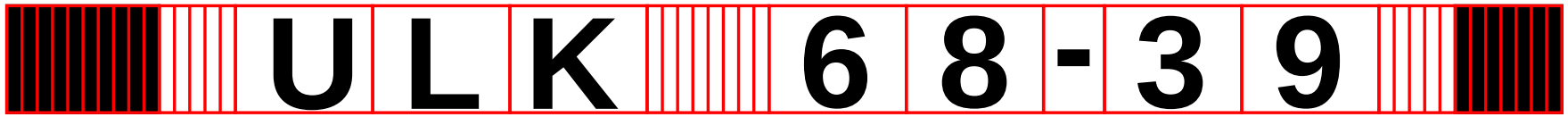


Segmentace a rozpoznávání textu

- Vstupní obrázek $\boldsymbol{x} \in \mathcal{X}$ můžeme segmentovat, tak že ze všech přípustných syntetických obrázků vybereme ten nejpodobnější, tj. ten s největší korelací:

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}} f(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$$

- Z výstupu klasifikátoru $\hat{\boldsymbol{y}} = (\hat{s}_1, \ldots, \hat{s}_L)$ získáme jak pozici znaků $k(\hat{s}_i)$, tak jejich jména $a(\hat{s}_i)$:



Segmentace a rozpoznávání textu

- Maximalizační úloha pořebná k ohodnocení klasifikátoru

$$\max_{\boldsymbol{y}\in\mathcal{Y}} \left[\sum_{i=1}^{L(y)} \sum_{j=1}^{\omega(a(s_i))} \langle \mathrm{col}(\boldsymbol{x}, j + k(s_i) - 1), \mathrm{col}(\boldsymbol{\theta}_{a(s_i)}, j) \rangle \right]$$

  se dá řešit efektivně pomocí dynamického programování.

- Pro $i = 1$ do W spočti

$$f_i = \max_{a\in\mathcal{A},\omega(a)\geq i} \left[\sum_{j=1}^{\omega(a)} \langle \mathrm{col}(\boldsymbol{x}, i - \omega(a) + j), \mathrm{col}(\boldsymbol{\theta}_a, j) \rangle + f_{i-\omega(a)} \right]$$

- Složitost úměrná $W \cdot \omega \cdot |\mathcal{A}|$

Učení klasifikátoru z příkladů

- Jak nejlépe určit $\boldsymbol{\theta} = \{\boldsymbol{\theta}_a \in \mathbb{R}^{H \times \omega(a)} | a \in \mathcal{A}\}$?

- Klasifikátor je instancí lineárního klasifikátoru

$$\begin{aligned} \hat{\boldsymbol{y}} &= \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}} f(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}) \\ &= \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}} \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}) \rangle \end{aligned}$$

  kde $\boldsymbol{\Psi} \colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^n$.

- Neznámé parametry $\boldsymbol{\theta}$ můžeme učit z příkladů

$$\{(\boldsymbol{x}^1, \boldsymbol{y}^1), \ldots, (\boldsymbol{x}^m, \boldsymbol{y}^m)\} \in (\mathcal{X} \times \mathcal{Y})^m \, .$$

Učení klasifikátoru z příkladů

- **Formulace učení:** nalezni parametry $\boldsymbol{\theta} \in \mathbb{R}^n$ tak, aby klasifikátor segmentoval trénovací obrázky co nejpodobněji manuální anotaci.
- Zaveďme ztrátovou funkci $\Delta\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, tak že

$$\Delta(\boldsymbol{y}, \boldsymbol{y}') = \sum_{i=1}^{L(\boldsymbol{y})} [\![a(\boldsymbol{y}, i) \neq a(\boldsymbol{y}', i)]\!]$$

  kde $a\colon \mathcal{Y} \times \{1, \ldots, L\} \to \mathcal{A}$ vrací jméno znaku pokrývajícího i-tý sloupec dané segmentace.
- Učení lze formulovat jako optimalizační úlohu

$$\boldsymbol{\theta}^* = \operatorname*{argmin}_{\boldsymbol{\theta} \in \Theta} \frac{1}{m} \sum_{i=1}^{m} \Delta(\boldsymbol{y}^i, h(\boldsymbol{x}^i; \boldsymbol{\theta}))$$

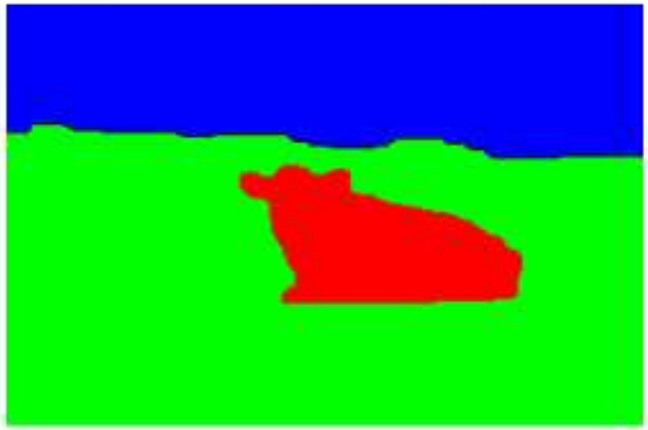
Segmentace a rozpoznávání textu

- Demo ukázka.
- Camea web service.

Aplikace IV: Segmentace obrazu

Segmentace obrazu

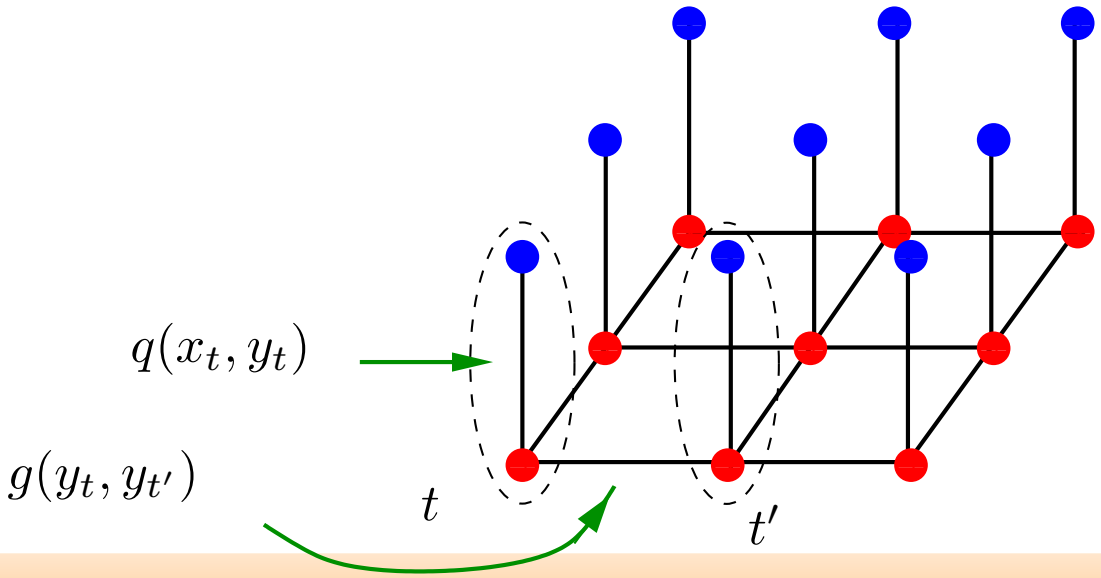
- **Cíl:** přiřadit každý pixel vstupního obrázku do jedné z K tříd.
- Typickým segmentační model: “max-sum” klasifikátor.



Segmentace pomocí max-sum klasifikátoru

- $\mathcal{T} = \{(i,j) \in \mathcal{N}^2 \mid 1 \leq i \leq W, 1 \leq j \leq H\}$ množina pixelů
- $\mathcal{E} = \{\{(i,j),(i',j')\} \in \mathcal{T}^2 \mid |i-i'| + |j-j'| = 1\}$ sousedící pixely
- $x_t = (r_t, g_t, b_t) \in X = \mathbb{R}^3$ barva pixelu $t \in \mathcal{T}$ v RGB prostoru
- $y_t \in Y = \{1, \ldots, K\}$ třída (jméno segmentu) pixelu $t \in \mathcal{T}$
- Homogenní model: $q_t(x_t, y_t) = q(x_t, y_t)$, $g_{tt'}(y_t, y_{t'}) = g(y_t, y_{t'})$



Segmentace pomocí max-sum klasifikátoru

- Klasifikátor najde segmentaci $\hat{\boldsymbol{y}}$, která nejlépe odpovídá vstupnímu obrázku $\boldsymbol{x}$ a zároveň splňuje apriorní předpoklady o segmentaci:

$$\hat{\boldsymbol{y}} = h(\boldsymbol{x};\boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{y}\in\mathcal{Y}} \Bigg[\underbrace{\sum_{t\in\mathcal{T}} q(x_t, y_t)}_{\textbf{shoda s obrázkem}} + \underbrace{\sum_{\{t,t'\}\in\mathcal{E}} g(y_t, y_{t'})}_{\textbf{apriorní znalost}} \Bigg]$$

- Funkce $q(x, y)$ učuje "pravděpodobnost", že pixel s barvou x patří do segmentu y.
- Funkce $g(y, y')$ učuje "pravděpodobnost", že sousední pixely budou patřit k segmentu y a y', např. $g(y, y) > g(y, y')$, $y \neq y'$ preferuje sousední pixely patřící stejnému segmentu, tj. vyžaduje kompaktní segmentace.

Jak určit funkce q a g ?

- Uvažujme lineární prametrizaci funkce q

$$q(x_t, y_t; \boldsymbol{\theta}) = \langle \boldsymbol{\theta}^g_{y_t}, \boldsymbol{\phi}^q(x_t) \rangle$$

kde $\boldsymbol{\phi}\colon \mathbb{R}^3 \to \mathbb{R}^p$ jsou příznaky spočtené z x_t, například

$$\boldsymbol{\phi}(x_t) = (r_t, g_t, b_t, r_t^2, g_t^2, b_t^2, 1)$$

- Uvažuje lineární parametrizci funkce g, např.

$$g(y_t, y_{t'}; \boldsymbol{\theta}) = \theta^g \cdot [\![y_t = y_{t'}]\!] = \begin{cases} \theta^g & \text{pokud} \quad y_t = y_{t'} \\ 0 & \text{pokud} \quad y_t \neq y_{t'} \end{cases}$$

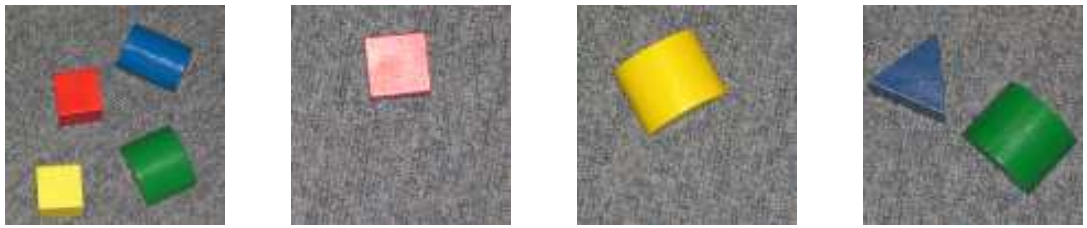
- *Poznámka: Pokud $\theta^g \geq 0$ a $|Y| = 2$ je funkce g submodulární a max-sum problem je efektivně řešitelný.*

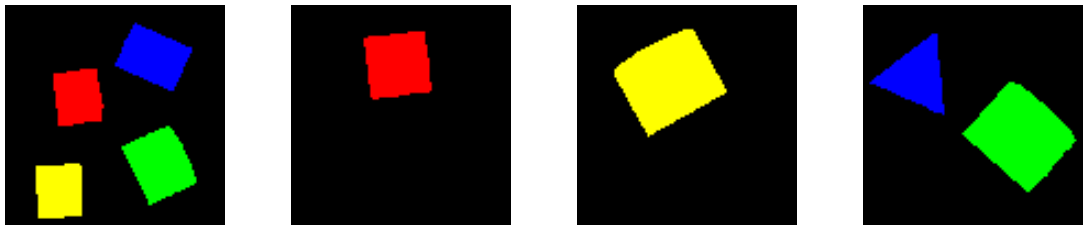
Učení max-sum klasifikátoru z příkladů

- **Úloha učení:** nalezni funkce $\boldsymbol{\theta} = (q, g)$ tak, tak aby max-sum klasifikátor měl na trénovací množině $\{(\boldsymbol{x}^1, \boldsymbol{y}^1), \ldots, (\boldsymbol{x}^m, \boldsymbol{y}^m)\}$ minimální chybu.
- Chybu můžeme měřit Hammingovou ztrátovou funkcí $\Delta \colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^n$

$$\Delta(\boldsymbol{y}, \boldsymbol{y}') = \sum_{t \in \mathcal{T}} [\![y_t \neq y_{t'}]\!]$$

- Příklad trénovací množiny:

$\{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^m\} =$ 

$\{\boldsymbol{y}^1, \ldots, \boldsymbol{y}^m\} =$ 

Učení max-sum klasifikátoru z příkladů

- Při lineární parametrizaci funkcí q_t a $g_{tt'}$ je max-sum klasifikátor instancí lineárního klasifikátoru

$$\begin{aligned} h(\boldsymbol{x};\boldsymbol{\theta}) &= \operatorname*{argmax}_{\boldsymbol{y}\in\mathcal{Y}} \left[\sum_{t\in\mathcal{T}} \langle \boldsymbol{\theta}, \boldsymbol{\Psi}^q(\boldsymbol{x}, \boldsymbol{y}_t) \rangle + \sum_{\{t,t'\}\in\mathcal{E}} \langle \boldsymbol{\theta}, \boldsymbol{\Psi}^g(\boldsymbol{y}_t, \boldsymbol{y}_{t'}) \rangle \right] \\ &= \operatorname*{argmax}_{\boldsymbol{y}\in\mathcal{Y}} \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}) \rangle \end{aligned}$$

- Učení klasifikátoru z trénovací množiny můžeme formulovat jako optimalizační problém

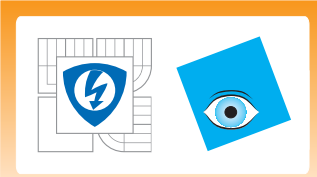
$$\boldsymbol{\theta} = \operatorname*{argmin}_{\boldsymbol{\theta}\in\Theta} \frac{1}{m} \sum_{i=1}^{m} \Delta(\boldsymbol{y}^i, h(\boldsymbol{x}^i;\boldsymbol{\theta}))$$

Příklad segmentace obrazu

- Příklad nestrukturního klasifikátoru: všechny pixely můžeme klasifikovat nezávisle pomocí nestrukturního klasifikátoru $h' : X \to Y$

$$h'(x_t) = \operatorname*{argmax}_{y \in Y} q(x_t, y) \,, \qquad t \in \mathcal{T}$$

- Matlab demo: rozdíl mezi strukturním a nestrukturním klasifikátorem.



Část II: Algoritmy učení

Plán přednášky

1. Formulace bayesovské klasifikační úlohy
2. Generativní a diskriminativní metody učení
3. Support Vector Machines
   - Algoritmus učení dvou-třídních lineárních klasifikátorů
4. Structured Output Support Vector Machines
   - Algoritmus učení strukturních lineárních klasifikátorů

Formulace bayesovské klasifikační úlohy

Formulace bayesovské klasifikační úlohy

- $\boldsymbol{x} \in \mathcal{X}$ je pozorovatelný stav (měření) a $\mathcal{X}$ je množina všech pozorování
- $\boldsymbol{y} \in \mathcal{Y}$ je skrytý stav ("label") a $\mathcal{Y}$ je množina všech skrytých stavů
- $p(\boldsymbol{x}, \boldsymbol{y})$ je h.p. generující pozorování a skryté stavy
- $h\colon \mathcal{X} \to \mathcal{Y}$ je klasifikátor
- $\Delta\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ je ztrátová funkce
- **Bayesovská úloha:** nalezni bayesovský klasifikátor, který minimalizuje očekávaný risk = střední hodnotu ztrátové funkce

$$R_{\text{exp}}(h) = \sum_{\boldsymbol{x} \in \mathcal{X}} \sum_{\boldsymbol{y} \in \mathcal{Y}} p(\boldsymbol{x}, \boldsymbol{y}) \Delta(\boldsymbol{y}, h(\boldsymbol{x}))$$

Obecnost bayesovské formulace

Aplikace	Pozorování $\boldsymbol{x} \in \mathcal{X}$	Skrytý stav $\boldsymbol{y} \in \mathcal{Y}$
Hodnota mince v automatu	$\boldsymbol{x} \in \mathbb{R}^n$	hodnota mince
Optical character recognition	2D šedotónový obrázek	text
Čtečka otisků	2D šedotónový obrázek	identita osoby
Rozpoznávání řeči	$x(t)$ zvukový signál	slova
EGG, ECG diagnostika	$x(t)$ časový signál	diagnóza
...	...	...

Aplikace, o kterých jsme mluvili

Sudoku solver	zadání hlavolamu	řešení hlavolamu
Segmentace obrazu	2D barevný obrázek	segmentace
Segment./rozpoz. textu	2D šedotónový obrázek	segmentace, znaky
Významné body na tváři	2D šedotónový obrázek	pozice bodů

Dva speciální případy bayesovských klasifikátorů

- Ve strukturním rozpoznávání je skrytý stav $\boldsymbol{y} \in \mathcal{Y}$ tvořen $|\mathcal{T}|$-ticí (pod)-stavů

$$\boldsymbol{y} = (y_t \in Y_t \mid t \in \mathcal{T})$$

- Dva důležité případy ztrátové funkce $\Delta\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ jsou:

1. 0/1-ztrátová funkce

$$\Delta(\boldsymbol{y}, \boldsymbol{y}') = [\![\boldsymbol{y} \neq \boldsymbol{y}']\!]$$

   Bayesovský risk $R_{\text{exp}}(h)$ je roven $P(h(\boldsymbol{x}) \neq \boldsymbol{y})$.

2. Hammingova vzdálenost

$$\Delta(\boldsymbol{y}, \boldsymbol{y}') = \sum_{t \in \mathcal{T}} [\![y_t \neq y'_t]\!]$$

   Rozšíření 0/1-ztrátové funkce pro strukturní klasifikátory.

Bayesův klasifikátor v případ 0/1-ztrátové funkce

- Očekávaný risk v případě 0/1-ztrátové funkce je roven

$$R_{\text{exp}}(h) = \sum_{\boldsymbol{x}\in\mathcal{X}} \sum_{\boldsymbol{y}\in\mathcal{Y}} p(\boldsymbol{x},\boldsymbol{y}) [\![\boldsymbol{y} \neq h(\boldsymbol{x})]\!]$$

  což je pravděpodobnost chybné klasifikace.

- Bayesovský klasifikátor minimalizující $R_{\text{exp}}(h)$ je roven

$$\begin{aligned} h(\boldsymbol{x}) &= \operatorname*{argmin}_{\boldsymbol{y}\in\mathcal{Y}} \sum_{\boldsymbol{y}'\in\mathcal{Y}} p(\boldsymbol{x},\boldsymbol{y}') [\![\boldsymbol{y}' \neq \boldsymbol{y}]\!] \\ &= \operatorname*{argmax}_{\boldsymbol{y}\in\mathcal{Y}} p(\boldsymbol{y} \mid \boldsymbol{x}) \end{aligned}$$

  vede na Maximal A Posteriori (MAP) odhad skrytého stavu.

Bayesův klasifikátor v případě Hammingovy ztrátové funkce

- Očekávaný risk v případě Hammingovy ztrátové funkce je

$$R_{\text{exp}}(h) = \sum_{\boldsymbol{x}\in\mathcal{X}} \sum_{\boldsymbol{y}\in\mathcal{Y}} p(\boldsymbol{x},\boldsymbol{y}) \sum_{t\in\mathcal{T}} [\![y_t \neq h_t(\boldsymbol{x})]\!]$$

  což je průměrný počet chybně klasifikovaných pod-stavů.
- Bayesovský klasifikátor minimalizující $R_{\text{exp}}(h)$ je roven

$$\begin{aligned} h(\boldsymbol{x}) &= \operatorname*{argmin}_{\boldsymbol{y}\in\mathcal{Y}} \sum_{\boldsymbol{y}'\in\mathcal{Y}} p(\boldsymbol{y}' \mid \boldsymbol{x}) \sum_{t\in\mathcal{T}} [\![y'_t \neq \boldsymbol{y}]\!] \\ &= \left(h_1(\boldsymbol{x}), \ldots, h_{|\mathcal{T}|}(\boldsymbol{x}) \right) \end{aligned}$$

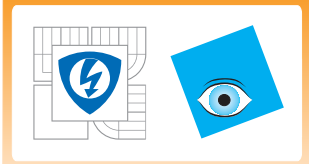
  kde $h_t(\boldsymbol{x}) = \operatorname{argmax}_{y\in Y_t} p(y \mid \boldsymbol{x})$

Řešení bayesovské úlohy

- K vyřešení potřebujeme znát h.p. $p(\boldsymbol{x}, \boldsymbol{y})$, resp. $p(\boldsymbol{y} \mid \boldsymbol{x})$.
- V praxi téměř vždy platí:
  - Úplný statistický model není k dispozici.
  - Problém můžeme popsat empirickými daty

$$\mathcal{D} = \{(\boldsymbol{x}^1, \boldsymbol{y}^1), \ldots, (\boldsymbol{x}^m, \boldsymbol{y}^m)\} \in (\mathcal{X} \times \mathcal{Y})^m$$

o kterých se předpokládá, že jsou realizací nezávislých náhodných proměnných se sdruženou h.p. $p(\boldsymbol{x}, \boldsymbol{y})$.



Generativní a diskriminativní metody učení

Generativní a diskriminativní metody učení

- **Generativní přístup:**
  - Definuj množinu $\mathcal{P}$, která obsahuje rozdělení podobná $p(\boldsymbol{x}, \boldsymbol{y})$.
  - Použij data $\mathcal{D}$ k odhadu $\hat{p} \in \mathcal{P}$, který co nejlépe aproximuje $p(\boldsymbol{x}, \boldsymbol{y})$.
  - Dosaď $\hat{p}(\boldsymbol{x}, \boldsymbol{y})$ do vzorce pro bayesovský klasifikátor (plug-in Bayes classifier).
- **Diskriminativní přístup:**
  - Definuj množinu $\mathcal{H}$, která obsahuje klasifikátory podobné bayesovskému.
  - Definuj funkci $\hat{R}(h)$, která dobře aproximuje bayesovský risk $R_{\text{exp}}(h)$ a lze ohodnoti z data $\mathcal{D}$.
  - Nalezni klasifikátor $h \in \mathcal{H}$, který minimalizuje $\hat{R}(h)$.

Generativní metody učení

- K odkadu $\hat{p} \in \mathcal{P}$ se typicky používá metoda maximální věrohodnosti.
- V případě, že data $\mathcal{D}$ jsou i.i.d., je věrohodnostní funkce

$$F(p') = \prod_{i=1}^{m} p'(\boldsymbol{x}^i, \boldsymbol{y}^i)$$

  rovna pravěpodobnosti, že model p' vygeneroval data $\mathcal{D}$.
- Maximálně věrohodný (ML) odhad modelu vede na optimalizační problém

$$\hat{p} = \operatorname*{argmax}_{p' \in \mathcal{P}} \prod_{i=1}^{m} p'(\boldsymbol{x}^i, \boldsymbol{y}^i)$$

- Plug-in bayesovský klasifikátor $\boldsymbol{h}(\boldsymbol{x}) = \operatorname{argmin}_{\boldsymbol{y} \in \mathcal{Y}} \sum_{\boldsymbol{y}' \in \mathcal{Y}} \hat{p}(\boldsymbol{y}' \mid \boldsymbol{x}) \Delta(\boldsymbol{y}', \boldsymbol{y})$

Generativní metody: klady a zápory

+ Obecný princip, který lze použít i v případě nekompletních dat.

- Učení z nekompletních dat je velká přednost speciálně ve strukturním rozpoznávání.
- Manuální anotace dat je časově i finančně náročná (viz. ukázkové aplikace).

-/+ Těžký optimalizační problém pro mnoho strukturních statistických modelů (např. RMF). Existují stochastické algoritmy hledající přibližné řešení.

- Vlastnosti ML odhadu dobře prozkoumány ve statisice pro případ $m \to \infty$; v praxi je dat konečný počet.

- Nepřesnost při specifikaci $\mathcal{P}$ může mít značný vliv na přesnost výsledného klasifikátoru $\Rightarrow$ problém musíme dost dobře znát než ho můžeme řešit.

Diskriminativní přístup

- Očekávaný risk $R_{\text{exp}}(h)$ se aproximuje empirickým riskem

$$R_{\text{emp}}(h) = \frac{1}{m} \sum_{i=1}^{m} \Delta(\boldsymbol{y}^i, h(\boldsymbol{x}^i)) \,,$$

  který měří přesnost klasifikátoru h na trénovacím vzorku $\mathcal{D}$.

- Učení vede na optimalizační problém

$$\hat{h} = \operatorname*{argmin}_{h \in \mathcal{H}} R_{\text{emp}}(h)$$

- Třídu klasifikátorů $\mathcal{H}$ je nutné kontrolovat (např. regularizací) aby nedošlo k přefitování.

Diskriminativní přístup: klady a zápory

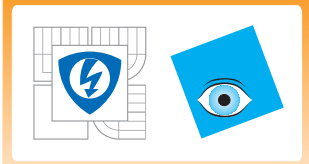
- Není (zatím) jasné jak zobecnit pro nekompletních data.

-/+ Těžký optimalizační problém, ale existují efektivní konvexní aproximace a to i pro složité strukturní modely jako max-sum klasifikátor (viz. dále).

+ Prozkoumán případ i pro konečné m.

- Teorie statistického učení zkoumá vztah $R_{\text{exp}}(h)$ a $R_{\text{emp}}(h)$; generalizační meze.
- Praktický výstup: při minimalizaci emprirického risku je třeba kontrolovat složitost třídy klasifikátorů $\mathcal{H}$ jinak dojde k přefitování.

+ Nepřesnost při specifikaci $\mathcal{H}$ nemá zničující vliv na přesnost výsledného klasifikátoru $\Rightarrow$ často lze použít jako “black-box approach”.



Implementace diskriminativního učení: Support Vector Machines

I. Algoritmus učení dvou-třídních lineárních klasifikátorů

Dvoutřídní lineární klasifikátor

- Pozorování je n-dimenzionální vektor $\boldsymbol{x} \in \mathcal{X} = \mathbb{R}^n$.
- Skrytý stav nabývá jen dvou hodnot $y \in \mathcal{Y} = \{+1, -1\}$
- Budeme učit lineární klasifikátor

$$h(\boldsymbol{x}; \boldsymbol{\theta}) = \begin{cases} +1 & \text{pokud} \quad \langle \boldsymbol{\theta}, \boldsymbol{x} \rangle \geq 0 \\ -1 & \text{pokud} \quad \langle \boldsymbol{\theta}, \boldsymbol{x} \rangle < 0 \end{cases}$$

  Pokud chceme klasifikátor s posunutím, pak použijeme $\boldsymbol{\theta}' = (\boldsymbol{\theta}; b)$ a $\boldsymbol{x}' = (x; 1)$.
- Uvažujeme 0/1-ztrátovou funkci (tj. chceme minimalizovat pravděpodobnost chybé klasifikace)

$$\Delta(y, y') = [\![y \neq y']\!]$$

Dvoutřídní lineární klasifikátor: separovatelná data

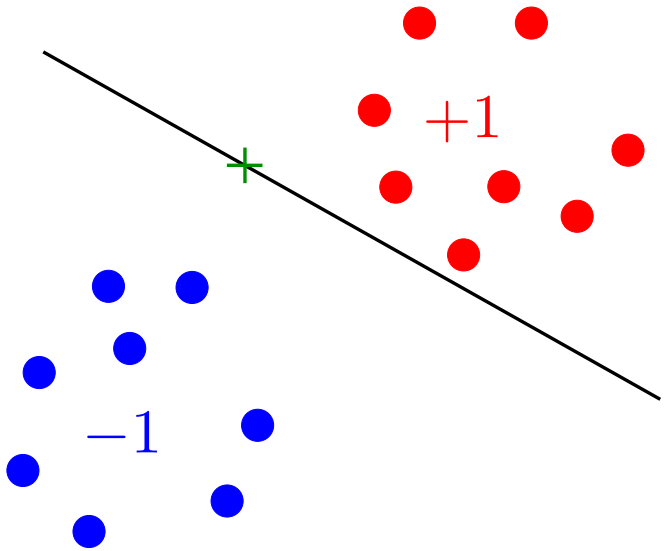
- Trénovací příklady $\mathcal{D} = \{(\boldsymbol{x}^1, y^1), \ldots, (\boldsymbol{x}^m, y^m)\} \in (\mathbb{R}^n \times \{+1, -1\})^m$ jsou lineárně separabilní pokud existuje vektor $\boldsymbol{\theta} \in \mathbb{R}^n$ takový, že

$$R_{\text{emp}}(\boldsymbol{\theta}) = \frac{1}{m} \sum_{i=1}^{m} [\![y^i \neq h(\boldsymbol{x}^i; \boldsymbol{\theta})]\!]$$

- Hledání $\boldsymbol{\theta}$, při kterém je $R_{\text{emp}}(\boldsymbol{\theta}) = 0$ vede na řešení soustavy lineárních nerovnic:

$$y^i \langle \boldsymbol{\theta}, \boldsymbol{x}^i \rangle > 0\,, \quad i = 1, \ldots, m$$

- Úloha LP řešitelná Perceptronem.



Optimální rozdělující nadpolocha

- Optimální rozdělující nadplocha $\langle \boldsymbol{\theta}^*, \boldsymbol{w} \rangle$ maximaluzuje vzdálenost od dat

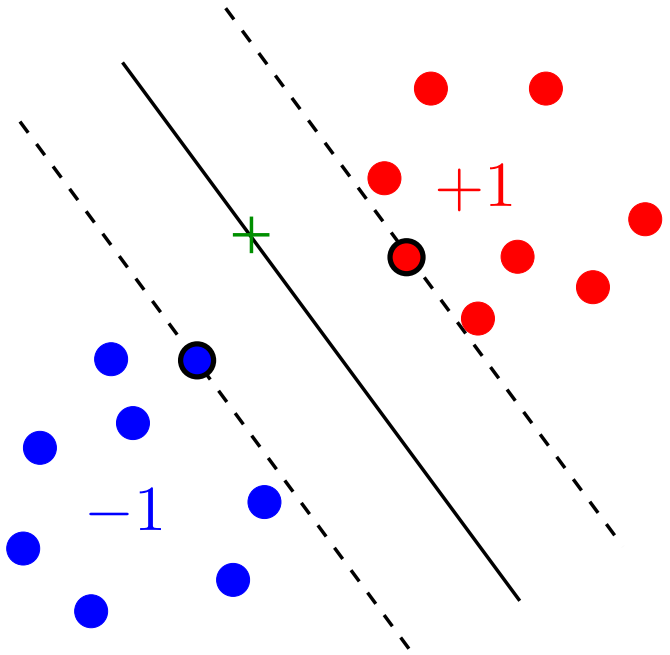
$$\boldsymbol{\theta}^* \in \operatorname*{argmax}_{\boldsymbol{\theta} \in \mathbb{R}^n} \min_{i=1,\ldots,m} \frac{y^i \langle \boldsymbol{\theta}, \boldsymbol{x}^i \rangle}{\|\boldsymbol{\theta}\|}$$

- Hledání optimální rozdělující nadplochy vede na úlohu kvadratického programování:

$$\boldsymbol{\theta}^* = \operatorname*{argmin}_{\boldsymbol{\theta}} \frac{1}{2} \|\boldsymbol{\theta}\|^2$$

za podmínky

$$y^i \langle \boldsymbol{\theta}, \boldsymbol{x}^i \rangle \geq 1 \,, \quad i = 1, \ldots, m$$



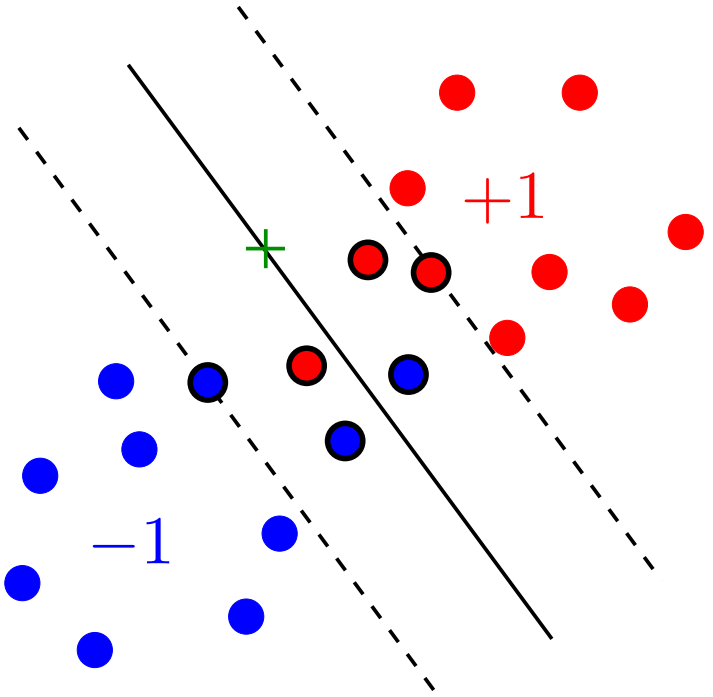
Lineární Support Vector Machine klasifikátor učený z neseparovatelných dat

- Lineární podmínky se uvolní pomocí relaxačních proměnných

$$(\boldsymbol{\theta}^*, \boldsymbol{\xi}^*) = \underset{\boldsymbol{\theta},\boldsymbol{\xi}}{\operatorname{argmin}} \left(\frac{\lambda}{2}\|\boldsymbol{\theta}\|^2 + \frac{1}{m}\sum_{i=1}^{m} \xi_i \right)$$

za podmínky

$$\begin{aligned} y^i \langle \boldsymbol{\theta}, \boldsymbol{x}^i \rangle &\geq 1 - \xi_i\,, \quad i = 1, \ldots, m \\ \xi_i &\geq 0\,, \quad i = 1, \ldots, m \end{aligned}$$



- $\lambda > 0$ je regularizační konstanta.
- Učení převedeno na problém konvexního kvadratického programování.

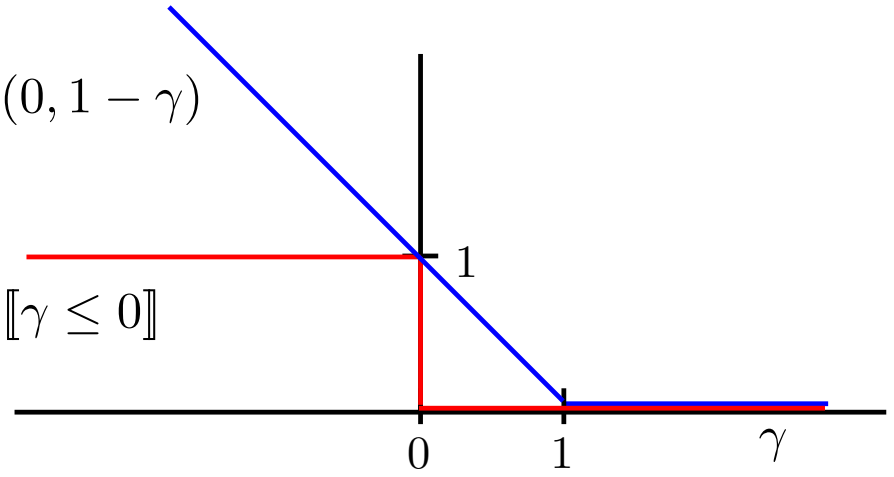
SVM implementuje princip minimalizace regularizovaného empirického risku

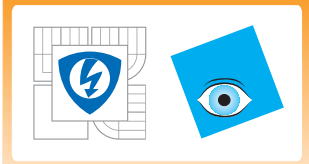
- SVM problém lze vyjádřit jako ekvivalentní problém bez omezení

$$\boldsymbol{\theta}^* = \operatorname*{argmin}_{\boldsymbol{\theta}} \left(\frac{\lambda}{2}\|\boldsymbol{\theta}\|^2 + R(\boldsymbol{\theta}) \right) \quad \text{kde} \quad R(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m} \max\left\{0, 1 - y^i \langle \boldsymbol{\theta}, \boldsymbol{x}^i \rangle\right\}$$

- SVM používá "hinge-loss", což je konvexní aproximace 0/1-ztrátové funkce

$$\begin{aligned} [\![y^i \neq h(\boldsymbol{x}^i; \boldsymbol{\theta})]\!] &= [\![y^i \langle \boldsymbol{\theta}, \boldsymbol{x}^i \rangle \leq 0]\!] \\ &\leq \max\{0, 1 - y^i \langle \boldsymbol{\theta}, \boldsymbol{x}^i \rangle\} \end{aligned}$$





Implementace diskriminativního učení: Structured Output Support Vector Machines

II. Učení obecného (strukturní) lineárního klasifikátoru

Obecný lineární klasifikátor

- Pozorování je $\boldsymbol{x} \in \mathcal{X}$
- Skrytý stav je $\boldsymbol{y} \in \mathcal{Y}$
- Budeme učit lineární klasifikátor

$$h(\boldsymbol{x}; \boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}} \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}) \rangle$$

  kde $\boldsymbol{\Psi} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^n$ je dané zobrazení z prostoru $\mathcal{X} \times \mathcal{Y}$ na prostor parametrů $\mathbb{R}^n$.

- Ukázkové strukturní klasifikátory jsou instance obecného lineárního klasifikátoru.

Učení obecného lineárního klasifikátoru ze separabilních dat

- Trénovací příklady $\mathcal{D} = \{(\boldsymbol{x}^1, \boldsymbol{y}^1), \ldots, (\boldsymbol{x}^m, \boldsymbol{y}^m)\} \in (\mathcal{X} \times \mathcal{Y})^m$ jsou lineárně separabilní pokud existuje vektor parametrů $\boldsymbol{\theta} \in \mathbb{R}^n$ takový, že

$$R_{\text{emp}}(\boldsymbol{\theta}) = \frac{1}{m} \sum_{i=1}^{m} [\![\boldsymbol{y}^i \neq h(\boldsymbol{x}^i; \boldsymbol{\theta})]\!]$$

- Hledání $\boldsymbol{\theta}$, při kterém je $R_{\text{emp}}(\boldsymbol{\theta}) = 0$ vede na řešení soustavy lin. nerovnic:

$$\langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\boldsymbol{x}^i, \boldsymbol{y}^i) \rangle > \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\boldsymbol{x}^i, \boldsymbol{y}) \rangle \,, \quad i = 1, \ldots, m \,, \boldsymbol{y} \in \mathcal{Y} \setminus \{\boldsymbol{y}^i\}$$

- Počet omezujících podmínek je roven $m(|\mathcal{Y}| - 1)$, např. u segmentace obrazu je $|\mathcal{Y}| = K^{H \times W}$.
- Úloha LP řešitelná Perceptronem pokud umíme ohodnotit klasifikátor.

Structured output Perceptron

1. $\boldsymbol{\theta} = \mathbf{0}$
2. For $i = 1, \ldots, m$
   (a) Klasifikuj i-tý příklad

$$\hat{\boldsymbol{y}}_i = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}} \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\boldsymbol{x}^i, \boldsymbol{y}) \rangle$$

   (b) Pokud $\hat{\boldsymbol{y}}_i = \boldsymbol{y}_i$ nedělej nic jinak aktualizuj váhy

$$\boldsymbol{\theta} := \boldsymbol{\theta} + \boldsymbol{\Psi}(\boldsymbol{x}^i, \boldsymbol{y}^i) - \boldsymbol{\Psi}(\boldsymbol{x}^i, \hat{\boldsymbol{y}}^i)$$

3. Pokud se v kroku 2 neprovedla ani jedna aktualizace vah vrať vektor $\boldsymbol{\theta}$ a skonči. Jinak pokračuj krokem 2.

- Pokud jsou data lineárně separabilní najde v konečném počtu kroků řešení.

Učení obecného lineárního klasifikátoru z neseparabilní dat

- Chceme učit lineární klasifikátor

$$h(\boldsymbol{x};\boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{y}\in\mathcal{Y}} \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\boldsymbol{x},\boldsymbol{y})\rangle$$

  z trénovací množiny $\mathcal{D} = \{(\boldsymbol{x}^1,\boldsymbol{y}^1),\ldots,(\boldsymbol{x}^m,\boldsymbol{y}^m)\} \in (\mathcal{X}\times\mathcal{Y})^m$.

- Uvažujme libovolnou nezápornou ztrátovou funkci $\Delta\colon \mathcal{Y}\times\mathcal{Y}\to\mathbb{R}$

$$\Delta(\boldsymbol{y},\boldsymbol{y}') \geq 0\,, \qquad \boldsymbol{y}\in\mathcal{Y}, \boldsymbol{y}'\in\mathcal{Y}$$

- Minimalizace empirického risku $R_{\text{emp}}(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m}\Delta(\boldsymbol{y}^i, h(\boldsymbol{x};\boldsymbol{\theta}))$ je těžká nekonvexní optimalizační úloha.

- Můžeme nalézt konvexní aproximaci ztrátové funkce?

Konvexní aproximace ztrátové funkce

$$
\begin{aligned}
\Delta(\hat{\boldsymbol{y}}, h(\hat{\boldsymbol{x}}; \boldsymbol{\theta})) &= \max_{\boldsymbol{y} \in \mathcal{Y}} \Delta(\hat{\boldsymbol{y}}, \boldsymbol{y}) [\![\max_{\boldsymbol{y}' \in \mathcal{Y}} \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\hat{\boldsymbol{x}}, \boldsymbol{y}') \rangle - \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\hat{\boldsymbol{x}}, \boldsymbol{y}) \rangle \leq 0]\!] \\
&\leq \max_{\boldsymbol{y} \in \mathcal{Y}} \max \left\{ 0, \Delta(\boldsymbol{y}, \boldsymbol{y}) - \max_{\boldsymbol{y}' \in \mathcal{Y}} \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\hat{\boldsymbol{x}}, \boldsymbol{y}') \rangle + \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\hat{\boldsymbol{x}}, \boldsymbol{y}) \rangle \right\} \\
&= \max \left\{ 0, \max_{\boldsymbol{y} \in \mathcal{Y}} \left(\Delta(\hat{\boldsymbol{y}}, \boldsymbol{y}) + \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\hat{\boldsymbol{x}}, \boldsymbol{y}) \rangle \right) - \max_{\boldsymbol{y}' \in \mathcal{Y}} \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\hat{\boldsymbol{x}}, \boldsymbol{y}') \rangle \right\} \\
&\leq \max \left\{ 0, \max_{\boldsymbol{y} \in \mathcal{Y}} \left[\Delta(\hat{\boldsymbol{y}}, \boldsymbol{y}) + \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\hat{\boldsymbol{x}}, \boldsymbol{y}) \rangle \right] - \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \rangle \right\} \\
&= \underbrace{\max_{\boldsymbol{y} \in \mathcal{Y}} \left[\Delta(\hat{\boldsymbol{y}}, \boldsymbol{y}) + \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\hat{\boldsymbol{x}}, \boldsymbol{y}) \rangle \right] - \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \rangle}_{\text{konvexní horní mez ztrátové funkce}}
\end{aligned}
$$

Structured Output Support Vector Machine

- Učení obecného lineárního klasifikátoru vede na **konvexní minimalizační problém**

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}\in\mathbb{R}^n}{\operatorname{argmin}} \left[\frac{\lambda}{2}\|\boldsymbol{\theta}\|^2 + R(\boldsymbol{\theta})\right]$$

kde

$$R(\boldsymbol{\theta}) = \frac{1}{m}\sum_{i=1}^{m}\left[\max_{\boldsymbol{y}\in\mathcal{Y}}\left(\Delta(\boldsymbol{y}^i,\boldsymbol{y}) + \langle\boldsymbol{\theta},\boldsymbol{\Psi}(\boldsymbol{x}^i,\boldsymbol{y})\rangle\right) - \langle\boldsymbol{\theta},\boldsymbol{\Psi}(\boldsymbol{x}^i,\boldsymbol{y}^i)\rangle\right]$$

- Jak tento optimalizační prolém řešit?

Structured Output Support Vector Machine jako úloha kvadratického programování

- Učení SO-SVM klasifikátoru lze vyjádřit jako problém **kvadratického programování**

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}\in\mathbb{R}^n}{\operatorname{argmin}} \left[\frac{\lambda}{2}\|\boldsymbol{\theta}\|^2 + \frac{1}{m}\sum_{i=1}^{m}\xi_i\right]$$

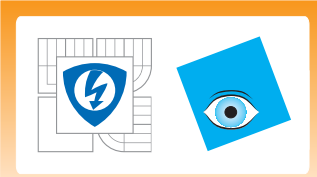
za podmínky

$$\xi_i \geq \Delta(\boldsymbol{y}^i, \boldsymbol{y}) + \langle\boldsymbol{\theta}, \boldsymbol{\Psi}(\boldsymbol{x}^i, \boldsymbol{y})\rangle - \langle\boldsymbol{\theta}, \boldsymbol{\Psi}(\boldsymbol{x}^i, \boldsymbol{y}^i)\rangle\,, \quad i = 1, \ldots, m\,, \boldsymbol{y} \in \mathcal{Y}$$

- Počet lineárních omezení je roven $m|\mathcal{Y}|$.
- Standardní algoritmy QP jsou nepoužitelné.

Literatura

- Česká učebnice:
  Schlesinger, Hlaváč. **Deset přednášek z teorie statistického a strukturního rozpoznávání.** *Vydavatelství ČVUT.* 1999.
- Statistická teorie učení, Support Vector Machines:
  Vapnik. **Statistical Learning Theory.** *John Wiley & Sons, Inc.* 1998.
- Původní práce o Structured Output Support Vector Machines:
  Tsochantaridis et. al. **Large Margin methods for structured and interdependent output variables**. *JMLR, 6:1453–1484.* 2005.
- Diskriminativní metody učení max-sum klasifikátorů:
  Franc, Savchynskyy. **Discriminative Learning of Max-Sum Classifiers.** *JMLR, 9, 67–104.* 2008.



Část III: Optimalizační algoritmy

Plán přednášky

1. Zobecnění gradientu pro nehladké funkce
2. Zobecněná gradientní metoda
3. Metody odsekávajících nadrovin (Cutting Plane Algorithm)

Zobecnění gradientu pro nehladké funkce

Formulace optimalizačního problému

- **Cíl: minimalizovat** kvadraticky regularizovanou **konvexní funkci**

$$\boldsymbol{\theta}^* = \operatorname*{argmin}_{\boldsymbol{\theta}\in\Theta} F(\boldsymbol{\theta}) := \left[\frac{\lambda}{2}\|\boldsymbol{\theta}\|^2 + R(\boldsymbol{\theta})\right]$$

  kde $R\colon \Theta \to \mathbb{R}$ je konvexní nehladká funkce.

- Pro jednoduchost budeme uvažovat případ bez omezení $\Theta = \mathbb{R}^n$.

- Minimalizační úloha je těžká kvůli risku $R(\boldsymbol{\theta})$: nedifferencovatelná funkce, samotné její ohodnocení je výpočetně náročné.

- Seznam algoritmů strojového učení, které vedou na regularizovanou konvexní minimalitaci je dlouhý. SVM algoritmy tvoří jen jeden příklad.

Subgradient (zobecněný gradient)

- Nechť $F : \Theta \to \mathbb{R}$ je **konvexní funkce** na konvexní množině $\Theta \subseteq \mathbb{R}^n$.
- Pokud je funkce F **diferencovatelná** v bodě $\boldsymbol{\theta}' \in \Theta$, **existuje gradient** $F'(\boldsymbol{\theta}') \in \mathbb{R}^n$, který určuje lineární dolní mez

$$F(\boldsymbol{\theta}) \geq F(\boldsymbol{\theta}') + \langle F'(\boldsymbol{\theta}'), \boldsymbol{\theta} - \boldsymbol{\theta}' \rangle , \quad \forall \boldsymbol{\theta} \in \Theta$$

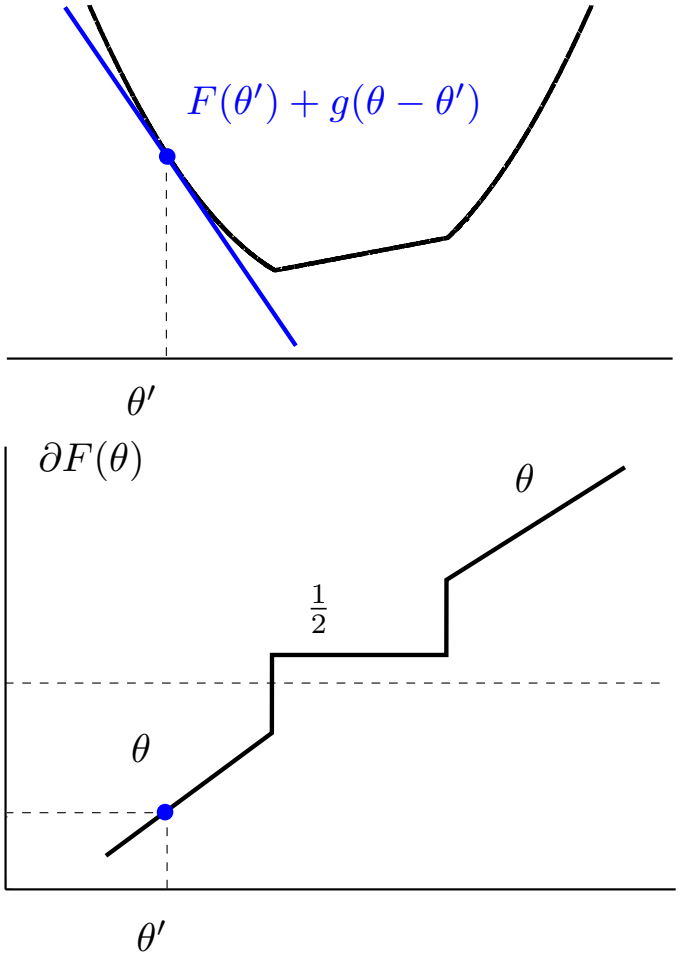
- Pokud je funkce F **nediferencovatelná** v bodě $\boldsymbol{\theta}' \in \Theta$, přesto musí existovat vektor $\boldsymbol{g} \in \mathbb{R}^n$ takový, že

$$F(\boldsymbol{\theta}) \geq F(\boldsymbol{\theta}') + \langle \boldsymbol{g}, \boldsymbol{\theta} - \boldsymbol{\theta}' \rangle , \quad \forall \boldsymbol{\theta} \in \Theta$$

  Vektor $\boldsymbol{g}$ je **subgradient funkce** F v bodě $\boldsymbol{\theta}$.
- V daném bodě může být více jak jeden subgradient. Množina $\partial F(\boldsymbol{\theta})$ všech subgradientů funkce F v bodě $\boldsymbol{\theta} \in \Theta$ se nazývá **subdifferenciál.**

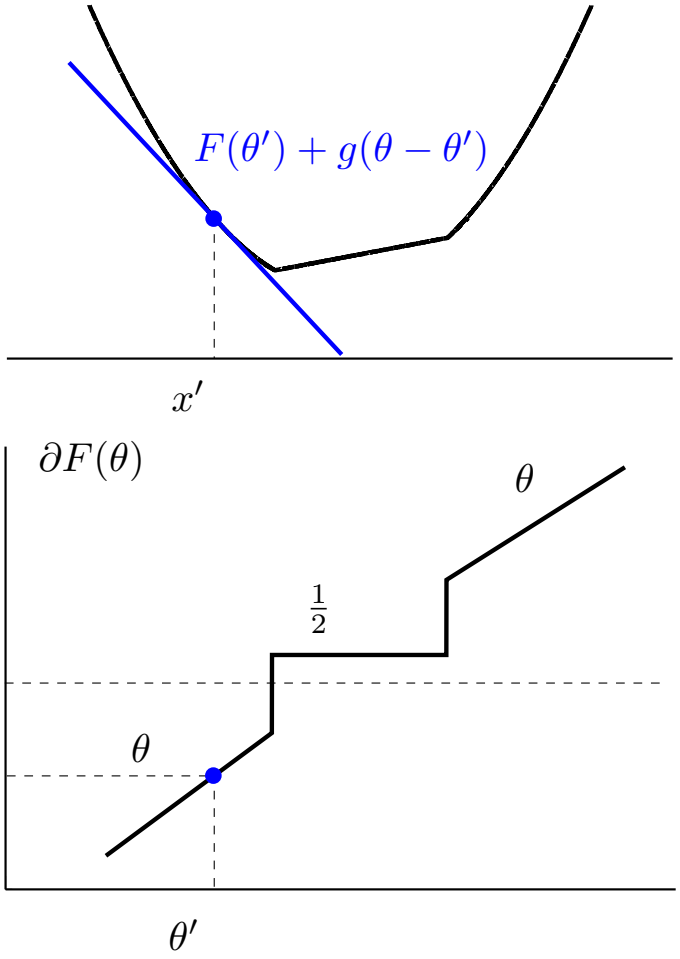
Příklad

$$F(\theta) = \max\{\tfrac{1}{2}\theta^2, \tfrac{1}{2}\theta + 2\}$$



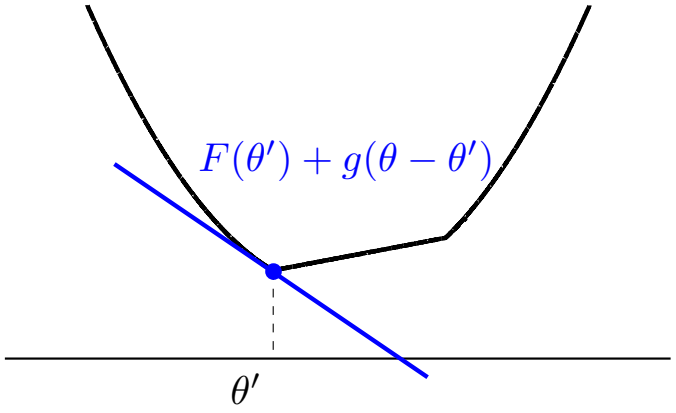
Příklad

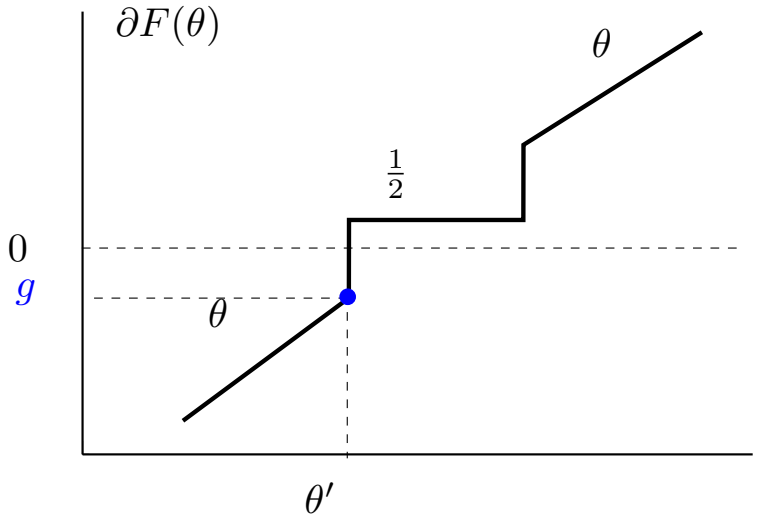
$$F(\theta) = \max\{\tfrac{1}{2}\theta^2, \tfrac{1}{2}\theta + 2\}$$



Příklad

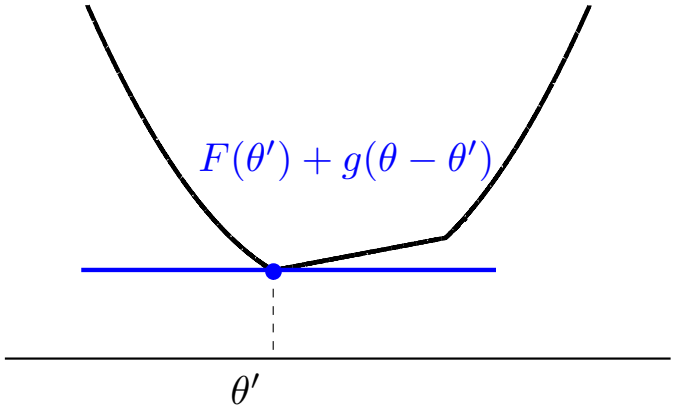
$$F(\theta) = \max\{\tfrac{1}{2}\theta^2, \tfrac{1}{2}\theta + 2\}$$

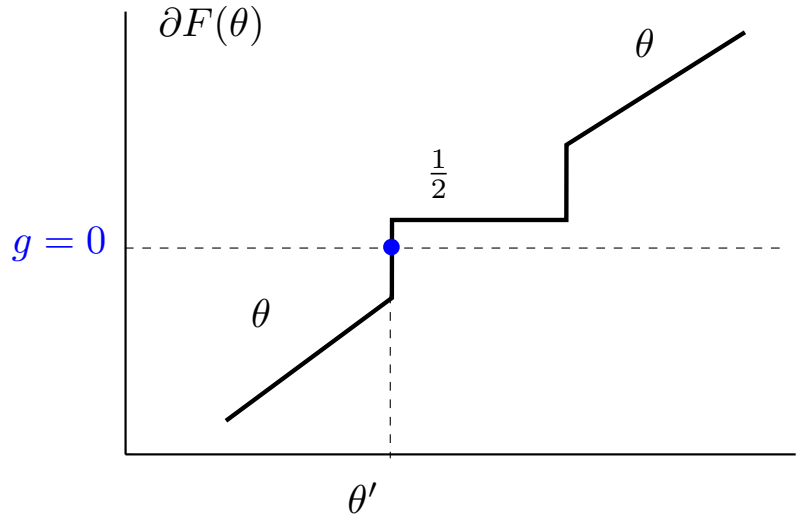




Příklad

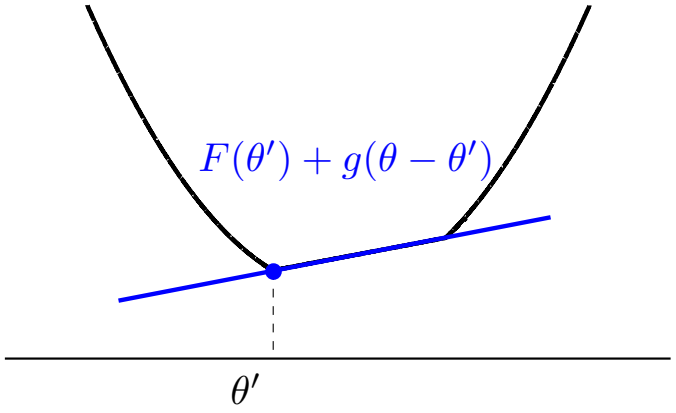
$$F(\theta) = \max\{\tfrac{1}{2}\theta^2, \tfrac{1}{2}\theta + 2\}$$

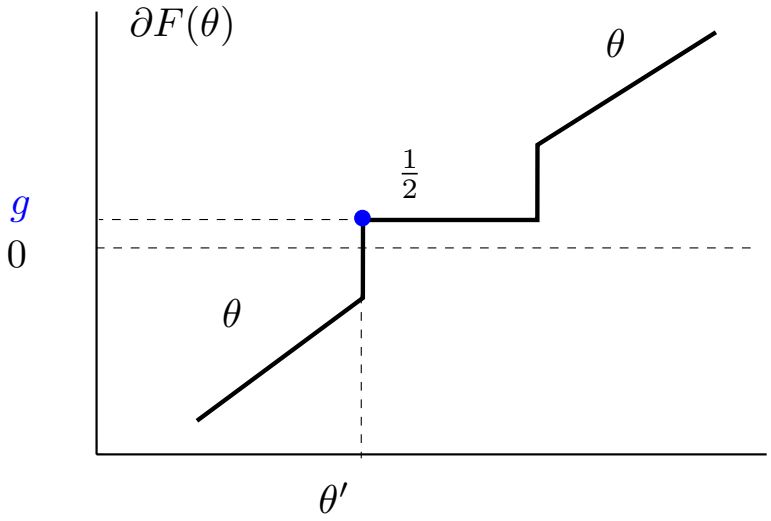




Příklad

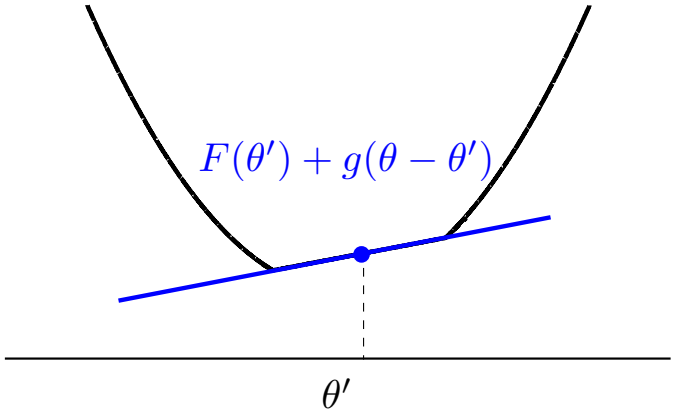
$$F(\theta) = \max\{\tfrac{1}{2}\theta^2, \tfrac{1}{2}\theta + 2\}$$

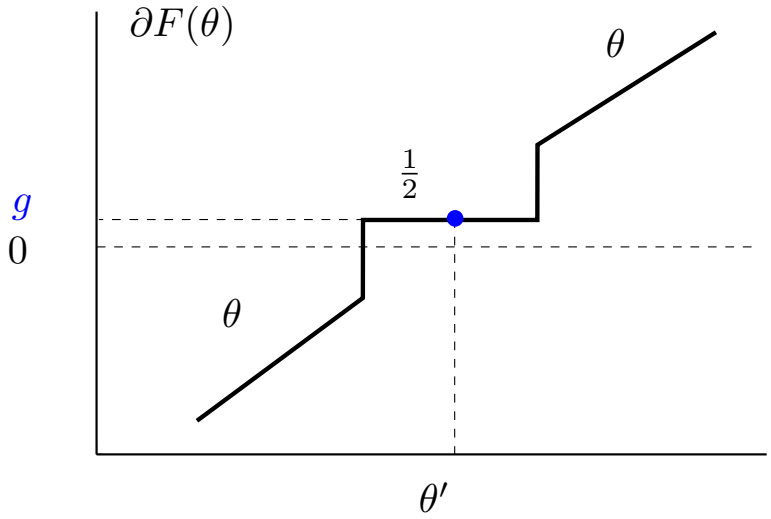




Příklad

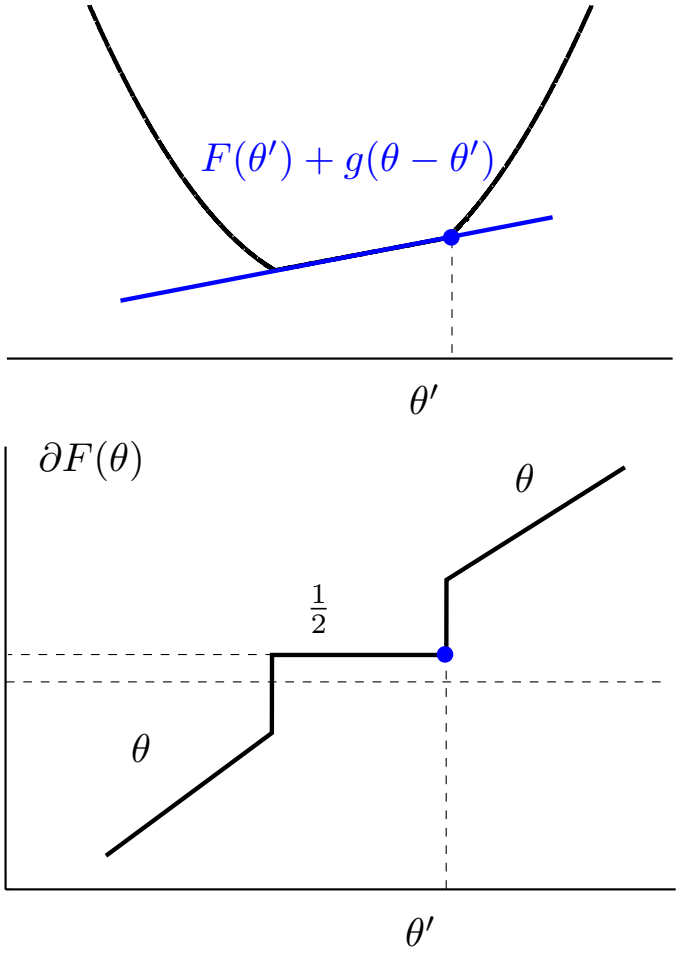
$$F(\theta) = \max\{\tfrac{1}{2}\theta^2, \tfrac{1}{2}\theta + 2\}$$





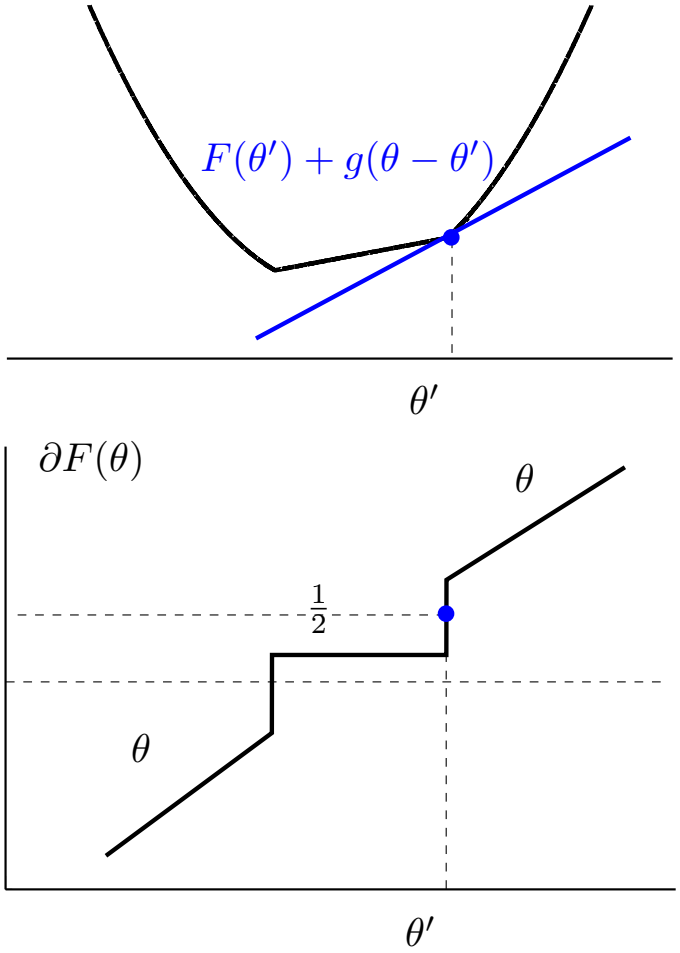
Příklad

$$F(\theta) = \max\{\tfrac{1}{2}\theta^2, \tfrac{1}{2}\theta + 2\}$$



Příklad

$$F(\theta) = \max\{\tfrac{1}{2}\theta^2, \tfrac{1}{2}\theta + 2\}$$



Základní vlastnosti subgradientu

- $\partial F(\boldsymbol{\theta})$ je neprázdná pokud je F konvexní a konečná blízko $\boldsymbol{\theta}$
- $\partial F(\boldsymbol{\theta})$ je uzavřená konvexní množina
- $\partial F(\boldsymbol{\theta}) = \{\boldsymbol{g}\} \iff F$ je differencovatelná a $g = F'(\boldsymbol{\theta})$
- $\partial(\alpha F(\boldsymbol{\theta})) = \alpha \partial F(\boldsymbol{\theta})$ pokud $\alpha > 0$
- $\partial(F_1(\boldsymbol{\theta}) + F_2(\boldsymbol{\theta})) = \partial F_1(\boldsymbol{\theta}) + \partial F_2(\boldsymbol{\theta})$
- Bodové maximum: $F(\boldsymbol{\theta}) = \max_{i=1\ldots,n} F_i(\boldsymbol{\theta})$; pokud jsou $F_i(\boldsymbol{\theta})$ diferencovatelné, pak

$$\partial F(\boldsymbol{\theta}) = \mathrm{Co}\{F_i'(\boldsymbol{\theta}) \mid F_i(\boldsymbol{\theta}) = F(\boldsymbol{\theta})\}$$

tj. konvexní obal gradientů aktivních funkcí v bodě $\boldsymbol{\theta}$.

Zobecněná gradientní metoda

Zobecněná gradientní metoda

- Nechť $F : \mathbb{R}^n \to \mathbb{R}$ je konvexní funkce. Cílem je vyřešit

$$\boldsymbol{\theta}^* \in \operatorname*{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^n} F(\boldsymbol{\theta})$$

- **Algoritmus: Zobecněná gradientní metoda** (subgradient descend)

1. Nastav $t = 1$. Zvol $\boldsymbol{\theta}_t \in \mathbb{R}^n$ a řadu pozitivních kroků $\{\alpha_i\}_{i=1}^{\infty}$.
2. Spočti subgradient $\boldsymbol{g}_t \in \partial F(\boldsymbol{\theta}_t)$
3. Aktualizuj řešení
$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha_t \boldsymbol{g}_t$$
4. Pokud algoritmus nedokonvegoval, pak $t := t + 1$ a jdi na krok 2.

Vlastnosti zobecněné gradientní metody

+ **Jednoduchost a obecnost.** Stačí spočítat libovolný subgradient $\boldsymbol{g}_t \in \partial F(\boldsymbol{\theta}_t)$.

+ **Zaručena konvergence** pro vhodně zvolenou délku kroků.
  Příklad konvergentní délky kroku:

$$\alpha_t = \frac{1}{t}$$

- **Pomalá konvergence.** Podobně jako gradientní metody: na začátku konvergují rychle, ale v okolí optima pomalu.

- **Konvegence není monotóní.** $-\boldsymbol{g}_t$ nemusí určovat směr, ve kterém se F snižuje. Nelze použít "line-search".

- **Chybí ukončovací podmínka.**

Učení SO-SVM klasifikátorů pomocí subgradientní metody

- SO-SVM risk je nediferencovatelná konvexní funkce

$$R(\boldsymbol{\theta}) = \frac{1}{m} \sum_{i=1}^{m} \left[\max_{\boldsymbol{y} \in \mathcal{Y}} \left(\Delta(\boldsymbol{y}^i, \boldsymbol{y}) + \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\boldsymbol{x}^i, \boldsymbol{y}) \rangle \right) - \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\boldsymbol{x}^i, \boldsymbol{y}^i) \rangle \right]$$

- Použijeme: i) subgradient součtu dvou funkcí je součet subgradientů, ii) subgradient maxima funkcí, je gradient libovolné aktivní funkce.
- Subgradient $R(\boldsymbol{\theta})$ v bodě $\boldsymbol{\theta}$ je roven

$$\boldsymbol{g} = \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{\Psi}(\boldsymbol{x}^i, \hat{\boldsymbol{y}}^i) - \boldsymbol{\Psi}(\boldsymbol{x}^i, \boldsymbol{y}^i)$$

kde $\hat{\boldsymbol{y}}^i = \text{argmax}_{\boldsymbol{y} \in \mathcal{Y}} \left(\Delta(\boldsymbol{y}^i, \boldsymbol{y}) + \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\boldsymbol{x}^i, \boldsymbol{y}) \rangle \right)$

Učení SO-SVM klasifikátorů pomocí subgradientní metody

- Úloha učení je převedena na problém konvexní optimalizace

$$\boldsymbol{\theta}^* = \frac{\lambda}{2}\|\boldsymbol{\theta}\|^2 + R(\boldsymbol{\theta})$$

- **Algoritmus učení SO-SVM klasifikátorů:**
  1. Nastav $t = 1$. Zvol $\boldsymbol{\theta}_t \in \mathbb{R}^n$ a řadu pozitivních kroků $\{\alpha_t\}_{t=1}^{\infty}$.
  2. Klasifikuj všechny příklady modifikovaným lin. klasifikátorem

$$\hat{\boldsymbol{y}}^i = \underset{\boldsymbol{y}\in\mathcal{Y}}{\operatorname{argmax}} \left(\Delta(\boldsymbol{y}^i, \boldsymbol{y}) + \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\boldsymbol{x}^i, \boldsymbol{y}) \rangle \right), \qquad i = 1, \ldots, m$$

  3. Spočti subgradient $\boldsymbol{g}_t = \lambda\boldsymbol{\theta} + \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{\Psi}(\boldsymbol{x}^i, \hat{\boldsymbol{y}}^i) - \boldsymbol{\Psi}(\boldsymbol{x}^i, \boldsymbol{y}^i)$
  4. Aktualizuj řešení $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \alpha_t \boldsymbol{g}_t$
  5. Pokud algoritmus nedokonvegoval, pak $t := t + 1$ a jdi na krok 2.

Učení SO-SVM klasifikátorů pomocí subgradientní metody

- Obecný **lineární klasifikátor**

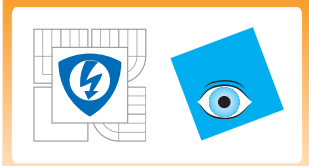
$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}} \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\boldsymbol{x}, y) \rangle$$

  můžeme **učit** pokud umíme **řešit modifikovanou klasifikační úlohu**

$$\hat{\boldsymbol{y}} = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}} \left(\Delta(\boldsymbol{y}', \boldsymbol{y}) + \langle \boldsymbol{\theta}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}) \rangle \right)$$

  Příklad: V případě max-sum klasifikátoru a hammingovy ztrátové funkce je modifikovaná klasifikační úloha stejně těžká jako normální klasif. úloha.

- Výhoda subgradientního algoritmu: **extrémně jednoduchá, obecná metoda.**
- Nevýhoda subgradientního algoritmu: **není jasné kdy ho ukončit.**



Metody odsekávajících nadrovin (Cutting Plane Algorithm)

Metody odsekávajících nadrovin (Cutting Plane Methods)

- **Hlavní myšlenka:** aproximovat složitou funkci $R(\boldsymbol{\theta})$ bodovým maximem množiny lineárních dolních mezí (cutting planes):

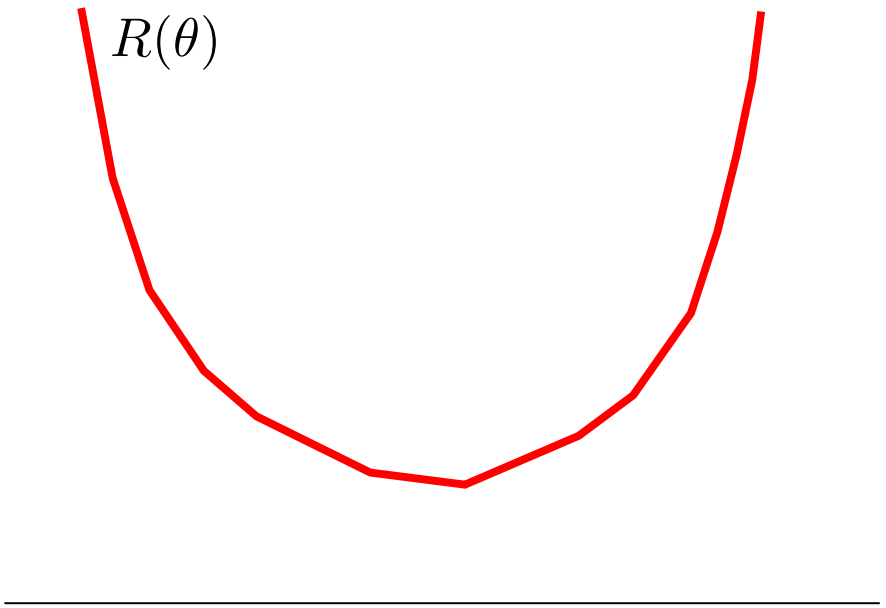
$$R_t(\boldsymbol{\theta}) = \max_{i=0,\ldots,t-1} \left[R(\boldsymbol{\theta}_i) + \langle \boldsymbol{g}_i, \boldsymbol{\theta} - \boldsymbol{\theta}_i \rangle \right]$$

  kde $\boldsymbol{g}_i \in \partial R(\boldsymbol{\theta}_i)$ je subgradient funkce F v bodě $\boldsymbol{\theta}_i$

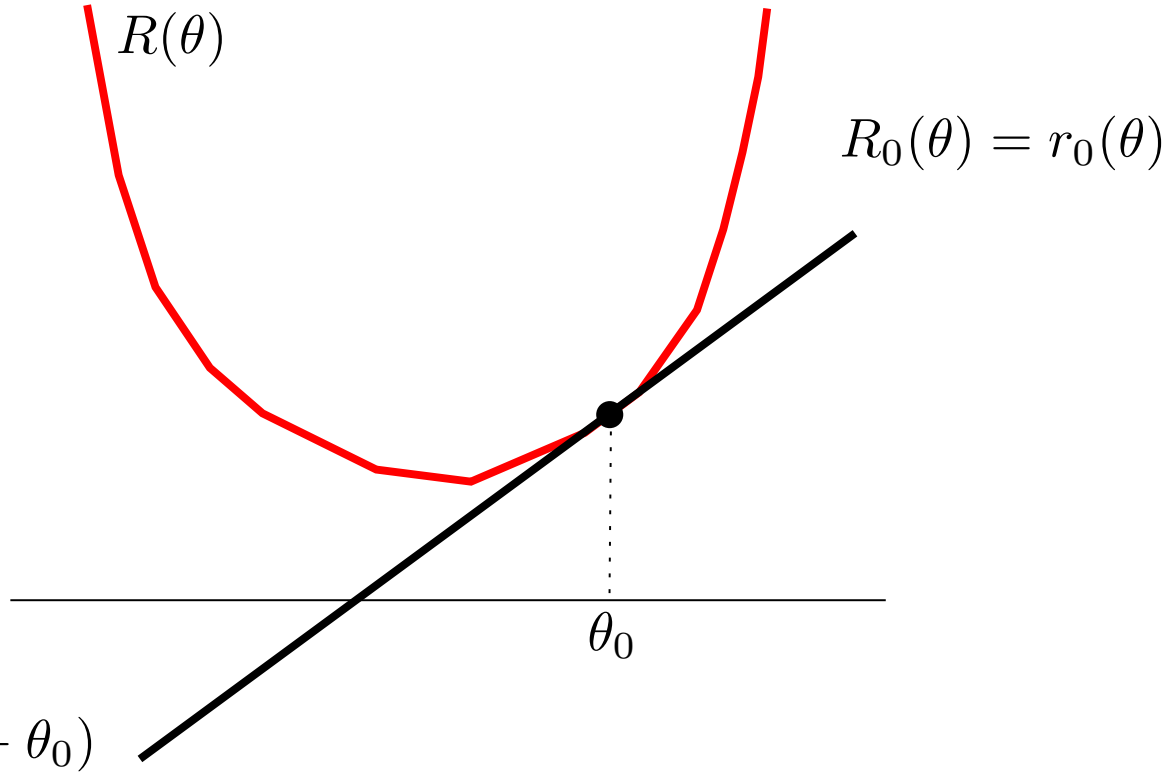
- **Cutting plane model** R_t je po částech lineární dolní mez funkce R, která je těsná v bodech $\boldsymbol{\theta}_i$, $i = 0, \ldots, t-1$, tj. platí:

$$\begin{aligned} R(\boldsymbol{\theta}) &\geq R_t(\boldsymbol{\theta}), \quad \forall \boldsymbol{\theta} \in \mathbb{R}^n \\ R(\boldsymbol{\theta}_i) &= R_t(\boldsymbol{\theta}_i), \quad i = 0, \ldots, t-1 \end{aligned}$$

Cutting Plane Model

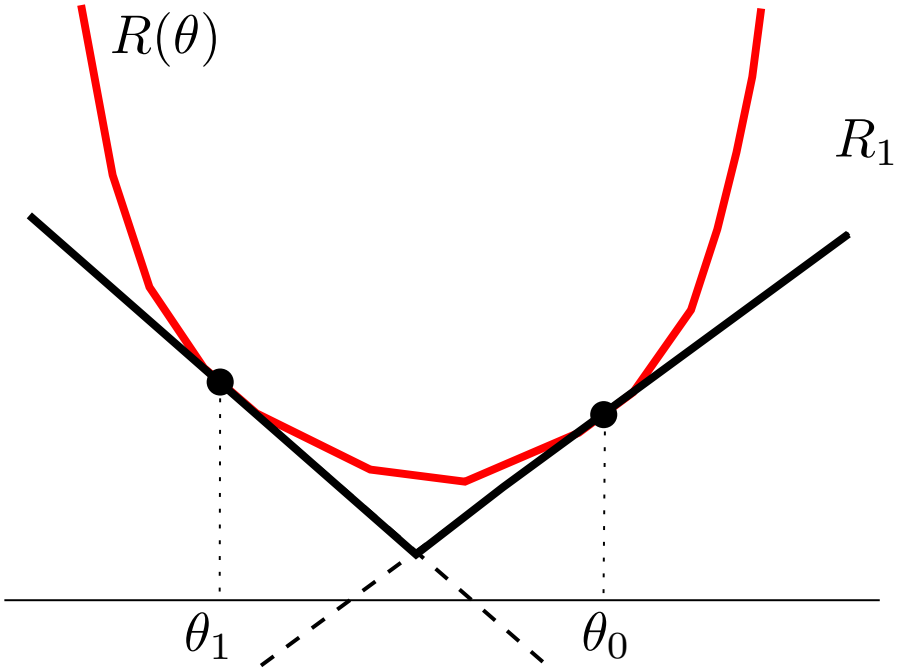


Cutting Plane Model



$$r_0(\theta) = R(\theta_0) + g_0(\theta - \theta_0)$$

Cutting Plane Model

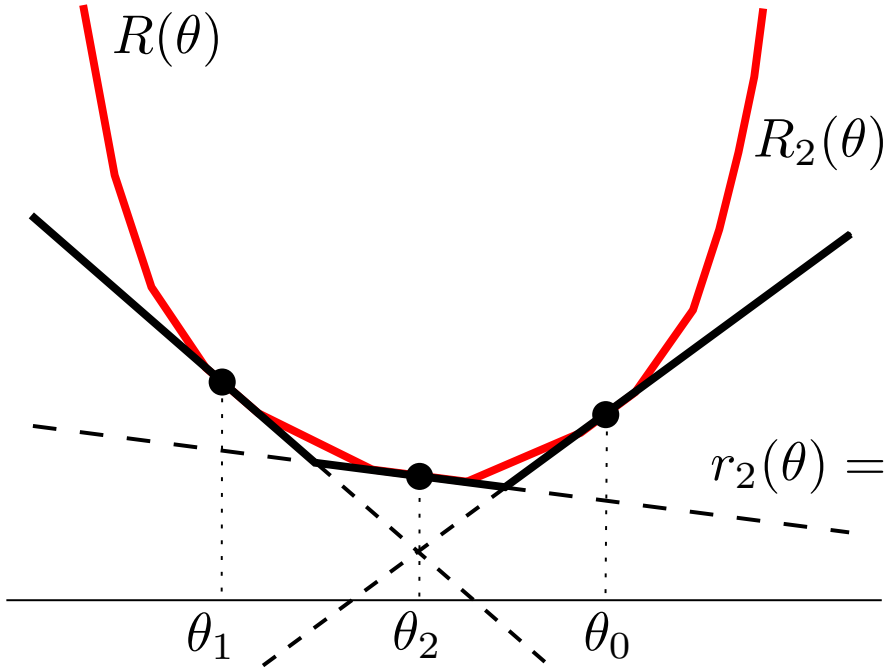


$$R_1(\theta) = \max\{r_0(\theta), r_1(\theta)\}$$

$$r_0(\theta) = R(\theta_0) + g_0(\theta - \theta_0)$$

$$r_1(\theta) = R(\theta_1) + g_1(\theta - \theta_1)$$

Cutting Plane Model



Cutting plane algoritmus minimalizující kvadraticky regularizovaný risk

- **Původní těžký konvexní problém**

$$\boldsymbol{\theta}^* = \operatorname*{argmin}_{\boldsymbol{\theta}\in\mathbb{R}^n} F(\boldsymbol{\theta}) := \left(\frac{\lambda}{2}\|\boldsymbol{\theta}\|^2 + R(\boldsymbol{\theta})\right)$$

se převede na **sekvenci jednoduchých redukovaných problémů**

$$\boldsymbol{\theta}_t = \operatorname*{argmin}_{\boldsymbol{\theta}\in\mathbb{R}^n} F_t(\boldsymbol{\theta}) := \left(\frac{\lambda}{2}\|\boldsymbol{\theta}\|^2 + R_t(\boldsymbol{\theta})\right)$$

kde $R_t(\boldsymbol{\theta})$ je "cutting plane model"

$$R_t(\boldsymbol{\theta}) = \max_{i=0,\ldots,t-1} \left[R(\boldsymbol{\theta}_i) + \langle \boldsymbol{g}_i, \boldsymbol{\theta} - \boldsymbol{\theta}_i\rangle\right]$$

složen z t lineárních členů (cutting planes).

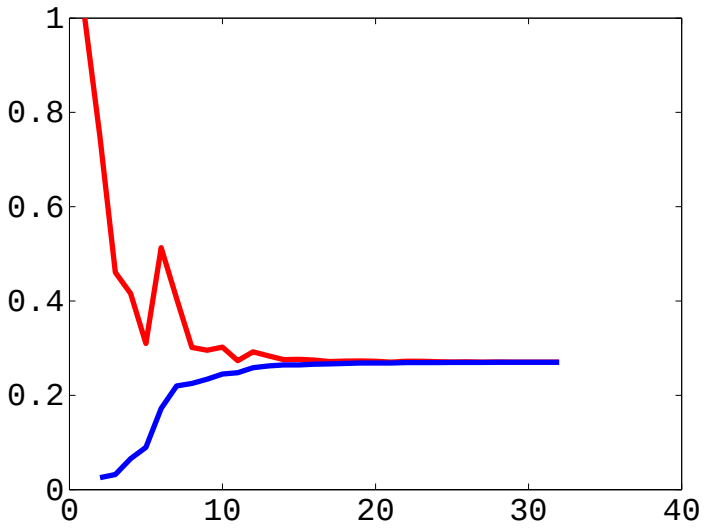
Cutting plane algoritmus

minimalizující kvadraticky regularizovaný risk

1. Inicializace: $t := 0$, $\boldsymbol{\theta}_t \in \mathbb{R}^n$
2. Spočti hodnotu risku $R(\boldsymbol{\theta}_t)$ a jeho subgradient $\boldsymbol{g}_t \in \partial R(\boldsymbol{\theta}_t)$
3. Aktualizuje cutting plane model přidáním $r_t(\boldsymbol{\theta}) = R(\boldsymbol{\theta}_t) + \langle \boldsymbol{g}_t, \boldsymbol{\theta} - \boldsymbol{\theta}_t \rangle$
4. Vyřeš redukovaný problém: $\boldsymbol{\theta}_{t+1} = \operatorname{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^n} \left(\frac{\lambda}{2} \|\boldsymbol{\theta}\|^2 + R_t(\boldsymbol{\theta}) \right)$
5. $t := t + 1$
6. Pokud $R(\boldsymbol{\theta}_t) - R_t(\boldsymbol{\theta}_t) \leq \varepsilon$ ukonči jinak pokračuj krokem 1.

$F(\boldsymbol{\theta}_t)$ červeně

$F_t(\boldsymbol{\theta}_t)$ modře



Jak řešit redukovaný problém

- Redukovaný problém lze převést na problém kvadratického programování

$$\boldsymbol{\theta}_t = \underset{\boldsymbol{\theta}\in\mathbb{R}^n,\xi\in\mathbb{R}}{\operatorname{argmin}} \left[\frac{\lambda}{2}\|\boldsymbol{\theta}\|^2 + \xi\right] \quad \text{s.t.} \quad \xi \geq \langle \boldsymbol{\theta}, \boldsymbol{g}_i \rangle + b_i \ , i = 0, \ldots, t-1$$

- Pokud máme hodně parametrů je vhodnější řešit duální problém

$$\boldsymbol{\alpha}_t = \underset{\boldsymbol{\alpha}\in\mathbb{R}^t}{\operatorname{argmin}} \left[\frac{1}{2\lambda}\langle \boldsymbol{\alpha}, \mathbf{A}^T\mathbf{A}\boldsymbol{\alpha}\rangle + \langle \boldsymbol{\alpha}, \boldsymbol{b}\rangle\right] \quad \text{s.t.} \quad \|\boldsymbol{\alpha}\|_1 = 1 \ , \boldsymbol{\alpha} \geq 0 \ .$$

  kdy platí, že $\boldsymbol{\theta}_t = -\lambda^{-1}\boldsymbol{A}\boldsymbol{\alpha}_t$.

- Duální problém má jednoduché omezující podmínky a jen t proměnných.
  Příklad: Detektor významných bodů má $\approx$ 10,000 parametrů.

Cutting plane algoritmus minimalizující kvadraticky regularizovaný risk

+ **Certifikát optimality:** řešení vrácené CPA se od optimáního neliší o více než ε, tj. platí

$$F(\boldsymbol{\theta}_t) \leq F(\boldsymbol{\theta}^*) + \varepsilon$$

+ **Zaručená konvergence:** pro libovolné $\varepsilon > 0$ algoritmus skončí né déle než za

$$T \leq \log_2 \frac{\lambda F(\mathbf{0})}{G^2} + \frac{4G^2}{\lambda\varepsilon} - 1$$

iterací, kde $G \geq \|\boldsymbol{g}\|$, $\forall \boldsymbol{g} \in \partial R(\boldsymbol{\theta}), \boldsymbol{\theta} \in \mathbb{R}^n$.

Příklad: V případě SO-SVM je G menší než průměr nejmenší koule obsahující všechny body $\boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y})$, $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{y} \in \mathcal{Y}$.

Cutting plane algoritmus minimalizující kvadraticky regularizovaný risk

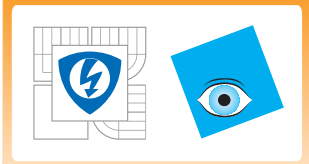
+ **Modularita:** aplikace na konkrétní úlohu učení vyžaduje pouze (first order oracle) proceduru počítající $R(\boldsymbol{\theta})$ a subgradient $\boldsymbol{g} \in \partial R(\boldsymbol{\theta})$.
  - Velká výhoda při návrhu nových algoritmů učení: pro každou modifikaci učícího problému není třeba navrhovat novou optimalizační metodu.

+ **Paralelizace:** časově dominantní částí algoritmu je výpočet $R(\boldsymbol{\theta})$ a $\boldsymbol{g} \in \partial R(\boldsymbol{\theta})$, který lze lehce počítat paralelně.

- **Pomalá konvergence** především při $\lambda \to 0$.
  - Pro $\lambda = 0$ se redukuje na čistý CPA o kterém je známo, že konverguje velmi pomalu.
  - V nejhorším případě je potřeba $m|\mathcal{Y}|$ iterací, po kterých $R(\boldsymbol{\theta}) = R_t(\boldsymbol{\theta})$ díky tomu, že $R(\boldsymbol{\theta})$ je sám bodové maximum lineárních funkcí.

Další rozšíření, literatura

- Původní publikace:
  Teo et al. **Bundle Methods for Regularized Risk Minimization**. *JMLR.* 11:311-365. 2010.
- Urychlená varianta CPA (line-search, lepší strategie výběru CP):
  Franc, Sonnenburg. **Optimized Cutting Plane Algorithm for Large-Scale Risk Minimization**. *JMLR*. 10:1532-4435. 2009.
- CPA má ve strojovém učení mnoho dalších aplikací:
  Franc, Sonnenburg, Werner. **Cutting-Plane Methods in Machine Learning**. Chapter in book *Optimization for Machine Learning*. MIT Press 2012.
- Lokalizační metody (konvergují rychle i při $\lambda = 0$):
  Antoniuk, Franc, Hlaváč: **Learning Markov Networks by Analytic Center Cutting Plane Method**. ICPR 2012.



Implementace

LIBQP: Knihovna kvadratického programování (ANSI C, Matlab).
`http://cmp.felk.cvut.cz/~xfrancv/libqp/html/index.html`

Shogun ML toolbox: Inference i učení strukturních klasifikátorů.
`http://www.shogun-toolbox.org/`

LIBOCAS: Urychlený CPA algoritmus pro učení SVM klasifikátorů.
`http://cmp.felk.cvut.cz/~xfrancv/ocas/html/index.html`