

RPZ - Main Vzorecky - sduzené pošt  $\rightarrow p_{xk}(x, k) = p_{xk}(x|k)p_k(k)$

1. Bayes, statistics: Strategie  $P(x=x \wedge K=k)$   $\rightarrow$  ztráta funkce

$$\text{Expected loss: } R(q) = \sum_{x \in X} \sum_{k \in K} p_{xk}(x, k) W(k, q(x))$$

pozorování ↑ P-trídy ↓ operační třída

$\rightarrow$  klasifikace podle strategie

Bayesova strategie  $q^*$ : minimalizuje  $R(q)$ .

Následme dělat pro každou  $x$  nezávisle:  $q^*(x) = \operatorname{argmin}_k \sum_{k' \in K} p_{k|x}(k|x) W(k, k')$ .

0-1 ztráta:  $q^*(x) = \operatorname{argmax}_{k \in K} p_{k|x}(k|x)$ .

Likelihood ratio: 2 třídy.  $q^*(x) = \operatorname{argmin}_k \left( \frac{p_{x11}(x)}{p_{x12}(x)} \right) / p_k(1) w(1, k) + p_k(2) w(2, k)$ .  $\rightarrow$  využití na tomto poměru lineární.

## 2. Nebayesovsky formulované slohy

### Neyman-Pearson

$K = \{D, N\}$  i známe  $p(x|D), p(x|N)$ ; neznáme  $p(D), p(N)$

dangerous normal

- zaručíme; chybu klasifikace pro  $D < \epsilon_D$ .

$\rightarrow$  a za této podmínky minimalizujeme chybu pro  $N$

Konstrukce strategie  $q^*$ :

Najdeme  $\mu$  minimální  $\mu$ , pro které platí podmínka  $\epsilon_D < \epsilon_D$  a pak:

$$\mu = \frac{p(x|N)}{p(x|D)} \quad x > \mu \Rightarrow q(x) = N$$

jinak  $q(x) = D$ .

S tabulkou: vytvoříme tabulku s  $\mu$

Minimax přidržíme od nejvýššího a kontrolojeme, aby  $N$  nepřeskočil hranu.

- známe stejné věci jako u Neyman Pearson, jen  $K = \{1, 2, \dots, N\}$ .

- minimalizujeme chybu nejhůře klasifikované třídy.

$$q^* = \operatorname{argmin}_q \max_{k \in K} \epsilon(k) \quad ; \quad \epsilon(k) = \sum_{x: q(x) \neq k} p(x|k)$$

- následně použijeme likelihood ratio (pro 2 třídy)

### Wald

$$\epsilon_1 \leq \epsilon; \epsilon_2 \leq \epsilon. \quad \epsilon_1 = \sum_{x: q(x)=1} p(x|1). \quad \epsilon_2 = \sum_{x: q(x)=2} p(x|2)$$

- specifikuje threshold na obou třídách,  $\epsilon_1, \epsilon_2$

- přidržíme možnost klasifikace "nevn".  $v_1, v_2 \rightarrow$  "undecided rate"

$q^* = \operatorname{argmin}_q \max_{x \in X} \psi_x$  za pod.:  $\epsilon_1 \leq \epsilon_1 \leq \epsilon_2 \leq \epsilon$  - rozsah  
 $(1) \text{ podle likelihood.}$

$$v_1 = \sum_{x: q(x)=1} p(x|1) - \mu_1 = \sum_{x: q(x)=1} p(x|2)$$

Fajné rozdělení a conjugate rozdělení k nim:

• Biomické rozdělení:

- výběr  $N$  vecí s opakováním z třídy.

$$P(R|N|\pi) = \binom{N}{R} \pi^R (1-\pi)^{N-R}$$

↳ jedna třída  $\sim$  procento třídy 1.  
normalizace  
aby byla p-st.

$$\hat{\pi}_{ML} = \frac{R}{N}$$

conjugate priori Beta distribuce

$$\beta(\pi|a, b) = \frac{1}{B(a, b)} \pi^{a-1} (1-\pi)^{b-1}$$

$$\hat{\pi}_{MAP} = \frac{R + A}{N + A + B}$$

$\rightarrow$  má A a B se  
možeme dívat jako na  
"fake" pozorování

$\sim$  normalizační  
člen (obsahuje gamma funkci)

• Normální rozdělení:

$$MLE: P(T|\mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{1}{2\sigma^2} \sum_{i=1}^N (x_i - \mu)^2}$$

$$\hat{\mu}_{ML} = \frac{1}{N} \sum x_i$$

$$\hat{\sigma}_{ML}^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \hat{\mu}_{ML})^2.$$

4. Neparametrické metody odhadu <sup>rozdělení</sup> ~~parametry~~ (histogramy, patzen, k-NN).

Histogram:  $\hat{d}_{\text{NUE}} = \frac{B N_E}{\sqrt{N}}$  → počet bále málezíach do  $i$ -tého binu  
↳ výška binu → celkový počet bins

$$d_{k,\text{MAP}} = B \frac{\frac{N_i + \lambda_i - 1}{N + \sum_{i=1}^B \lambda_i - B}}$$

selektion B: cross validace log-likelihoodu  $\ell = \sum_{j=1}^B N_j \log d_j$

### K-NN

Klasifikace - klasifikuj podle k nejbližších sousedů.

odhad rozdělení -  $p \sim \frac{1}{V}$ ,  $V$  je objem minimálního hyperkvalónu obsahujícího k-NN

Vlastnosti k-NN pro klasifikaci:

+ jednoduché

- pomalé, ale dá se zrychlit

- hodně parametrů, ale dá se zlepšit

- volba metriky (normalizace atd.)  $\nearrow$  Bayesovský error

- error bounds pro 1-NN:  $\varepsilon_B \leq \varepsilon_{1\text{NN}} \leq 2\varepsilon_B - \frac{B}{R-1} \varepsilon_B^2$

pro k-NN:  $\varepsilon_{k\text{NN}} \leq \varepsilon_B + \varepsilon_{1\text{NN}} / \sqrt{k \cdot \text{const}}$ .

Možné zrychlení/zlepšení:

- k-d tree (rychlejsí vyhledávání)

- odstoupení samplů, které nemají vliv

### Patzen (kernolová metoda)

## 5. Logistická regrese

- využívá log odds  $\ln\left(\frac{P(1|x)}{P(0|x)}\right)$ .

- v některých případech fótiž log odds zůvisí na x lineárně, resp.:

- normální rozdělení se stejnými kovariancemi

- nezávislé příznaky s binární klasifikací

- multinomialní naivní Bayes.

- můžeme tedy závislost modelovat na parametru  $w$ :

$$\ln\left(\frac{P(1|x)}{P(0|x)}\right) = w \cdot x \quad (\text{homogenní hyperplán, } s \\ \text{1 za konci } x \text{ a biasem za konci } w).$$

pokud třídí k  $\{0, 1\}$ , pak z konvergencí vyplývá:

$$p(k|x) = \frac{1}{1 + e^{-w \cdot x}} \quad \} \text{ sigmoida}$$

hledání  $w^*$ :  $w^* = \arg \max_w - \sum_{(x_k, t_k) \in T} \ln(1 + e^{-w \cdot x_k})$

$\underbrace{\quad}_{\text{log likelihood, také cross entropy}}$

$\rightarrow$  zde pak řešíme numerickým metody, resp. gradient descentem,

- navíc: funkce je konkávní, má tedy právě jedno minimum (konc. výpot, absolvované)

mohli minimalizovat.

pro více faktů:  $p(k|x) = \frac{e^{w_1 x_1}}{e^{w_1 x_1} + e^{w_2 x_2} + \dots + e^{w_k x_k}}$

## 6 - Perception

- přímý klasifikátor, nesmí se odhadovat p-sti, ale právě minimizuje empirický riz. (training error)

$$\hookrightarrow R_{\text{emp}}(g_\theta(x)) = \frac{1}{n} \sum_{i=1}^n w(g_\theta(x_i), t_i).$$

↪ počet dat.

⇒ můžeme  $R_{\text{emp}}$  neznamená můžeme expected risk  $R$  ⇒ (overfitting)

= dle se ospravedlnit kaprik a chet v orenkis rovnosti. → na tu všechno řešení.

- perceptron (obecně lineární klasifikátor) má VC dim =  $n+1$  (v  $n$ -rozměrném prostoru).

\* Lineární separabilita dat: data  $\in K \subseteq \{-1, 1\}$ .

$$\exists w \in \mathbb{R}^{D+1} : \text{sign}(w [x_i]) = k_i, \text{ pro } i=1, \dots, L.$$

## Klasifikace perceptronu:

(povídáme homogenní reprezentaci).

$$w \cdot \bar{x}_j \geq 0 \Rightarrow k_j = 1$$

$$w \cdot \bar{x}_j < 0 \Rightarrow k_j = -1.$$

ještě jednoduší podmínka když:  $x_j := k_j \cdot \bar{x}_j \Rightarrow$

## Algoritmus:

1. inicializuj  $w$  (pozorování:  $x_j = k_j \cdot \bar{x}_j \Rightarrow$  pak stačí zjistit, jestli  $w \cdot x_j > 0$ .)

2. najdi špatně klasif. pozorování

$$w \cdot x_j \leq 0$$

3. pokud není žádné  $\Rightarrow$  konec

jinak:  $w = w + x_j$ .

## Movikoff theorem

Pokud data jsou lineárně separovatelná a  $\exists w, \gamma \in \mathbb{R}^+$ :  $w \cdot x_j \geq \gamma \forall j$ ,  $\|w\|=1$ .

a  $D = \max_{x \in \mathcal{X}} \|x\|$ ; pak se perceptron zastaví po  $t^*$  krocích a

$$t^* \leq \frac{D^2}{\gamma^2}.$$

(6)

## 7. SVM

- maximalizují margin  $m = 2 \min_{x \in T} d(x)$ .
- > minimalizují také strukturní riziko (Lagrangian)

Původní formulace složky margin.

$$w^*, b^* = \underset{w, b}{\operatorname{argmax}} \min_{(x_i, y_i) \in T} d(x_i, y_i) \quad \text{za podmínek: } y_i(wx + b) > 1 \forall (x_i, y_i) \in T.$$

$$\text{a } d(x_i, y_i) = \frac{|y_i(wx + b)|}{\|w\|}.$$

$\rightarrow$  zde máme volné souřadnice  $\Rightarrow$  "fixujeme" - podmínka:  $y_i(wx + b) \geq 1$ . ( $a =$  pro support vectors).

- pak hledáme řešení bude:

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

převážitli jsou zlomky

$$\Rightarrow \text{vedle toho primární složky: } (w^*, b^*) = \underset{(w, b)}{\operatorname{argmin}} \frac{1}{2} \|w\|^2 \text{ a získat tak minimizaci.}$$

(QP - quadratic programming). za podmínek:  $y_i(wx + b) \geq 1 \forall (x_i, y_i) \in T$ .

$\Rightarrow$  dualní složka se nám dojít přes Lagrange a derivace podle  $w$  a  $b = 0$ .

$$d = \underset{\lambda}{\operatorname{argmax}} \left( \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i,j} \sum_{y_i y_j} \lambda_i \lambda_j y_i y_j x_i \cdot x_j \right).$$

za podmínek:  $\sum_i \lambda_i y_i = 0$  i  $\lambda_i \geq 0 \quad \forall i \in \{1, 2, \dots, N\}$ .

$$\text{pok: } w = \sum_{i=1}^n \lambda_i y_i x_i \quad b = y_s - w x_s \quad \text{pro jakékoli } (x_s, y_s) \text{ které je support vector.}$$

$$\Rightarrow w x + b = \sum_{i=1}^n \lambda_i y_i x_i^T x + b$$

$\rightarrow$  dual a primární složka splňují silnou dualitu.

Poz: data se vyskytují ve formě dot produktu  $x_i^T x_j$ .

$\rightarrow$  díl se využít promítání, nepotřebujeme vlastní transformaci  $\Phi(x)$ , stačí zadat pozitivně definovanou matice  $K = x^T K x$

$$K = \begin{matrix} & \text{semi} \\ \downarrow \text{kernel} \end{matrix}$$

## Soft-margins SUM

- povolíme chyby pomocí slack variables  $\xi_i$

→ pokud bod  $(x_i, \xi_i)$  poruší podmínku:  $\xi_i(w \cdot x_i + b) \geq 1$

→ relaxujeme  $\xi_i(w \cdot x_i + b) \geq 1 - \xi_i$  a platíme  $\xi_i \leq \epsilon_i$

$$\text{Prin.: } (w^*, b^*) = \underset{(w, b)}{\operatorname{argmin}} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i$$

novinka, the price we pay

$$\text{z.p.: } \xi_i(w \cdot x_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0 \quad \forall i = 1 \dots N$$

## Dual:

měníme pouze podmínku na multiplikátory:  $0 \leq \alpha_i \leq C$

## 8. ADABoost

⇒ máme set weak learners, které postupně stavíme a pak klasifikujeme jejich

- myslíme: (stromy, perceptrony) pro další klasifikátor v tuto zvyšuje vahu datum, které jsme špatně (combi mod.) klasifikovali

výsledek:  $D_t(x_1, \xi_1) \dots (x_L, \xi_L)$ ;  $\xi_i \in \{-1, 1\}$ .

Initializuj vahy dat:  $D_1(x) = \frac{1}{L} \rightarrow \text{rovnoměrné.}$

for  $t=1 \dots T \rightarrow$  index classifieru

- najdi funkci  $h_t$ , která klasifikuje nejméně soudit všechny okrajovky. n  
↳ to je ten weak classif.

pokud  $\xi_t \geq \frac{1}{2} \rightarrow$  konec -- moc špatných klasifikátorů

-  $\lambda_t = \frac{1}{2} \log \left( \frac{1-\xi_t}{\xi_t} \right)$  ( $\lambda_t > 0$ , protože  $\xi_t < \frac{1}{2}$ ). →  $\lambda_t$  je "váha" weak classifier

- aktualizujeme  $D_{t+1}(x) = \frac{D_t(x) e^{-\lambda_t \cdot h_t(x)}}{Z_t}$ .  $Z_t = \sum_{i=1}^L D_t(i) e^{-\lambda_t \cdot h_t(x_i)}$   
↳ pokud správná klasifikace máme  $e^{-\lambda_t}$   
 $e^{-\lambda_t} \rightarrow$  snížujeme.

## Klasifikace:

$$H(x) = \operatorname{sign} \left( \sum_{t=1}^T \lambda_t h_t(x) \right). \quad (8)$$

Upper bound AdaBoostu:  $\dots \rightarrow$  empirical error.

$$\text{error } \varepsilon = \frac{1}{L} \sum_i \varepsilon_i + H_T(x_i) \leq \prod_{t=1}^T Z_t$$

$\rightarrow$  AdaBoost minimalizuje tuto bound minimalizací  $Z_t$

Pozn.:  $Z_t$  a  $\varepsilon_t$  jsou odvozové právě z minimalizace  $Z_t$  (derivací)  
+ jednoduchý feature selection, strategie motivovaná minimalizací upper bound empir. <sup>add.</sup>

postupně zlepšujeme

- může overfitovat jednoduché (dá se na datu vysokouhu).

## 9. Neuronky.

Universal Approximation Theorem:  $\dots \times$  hyperkrychle.

$\rightarrow$  na omezené množině se pomocí lineární kombinace klasifikátorů

typu  $\sigma(u_i \cdot x + b_i)$  můžu dostat libovolně blízko správné klasifikaci  
( $\sigma$  linearita (sigmoid, tanh, RELU, sign))

### Neuronová síť:

$\rightarrow$  výstup, každá je funkce  $\mathbb{R}^n \rightarrow \mathbb{R}^m$  pro režimy  $n, m$ , na níž  
po složkách aplikujeme množinu  $\sigma$ .

$\rightarrow$  sigmoid není použitelné, chceme něco, co se dá derivovat.

- výsledek měříme pomocí nějaké kriterijní funkce

-> squared diff:  $\|f - g\|^2$ .

-> neg. log-likelihood:  $-g^T \log f$

$\rightarrow$  forward pass: oklasifikujeme data

$\rightarrow$  backward (backprop.): počítáme gradient od zadu

používáme řetízkové pravidlo:

$$\frac{\partial f(g(x))}{\partial x_k} = \sum_{i=1}^n \frac{\partial f}{\partial z_i} \frac{\partial z_i}{\partial x_k} = f' \cdot g'$$

$\hookrightarrow$  proto jde o protijeho směru.

(9)

jako by byla matici.  
(Jacobian)  $\rightarrow$  transponovaný

10. k-means: → používáme  $d(x_i, c_j) = \sqrt{\sum_{i=1}^n \|x_i - c_j\|^2}$ .
1. inicializace center  $c_k$  (nahodil, k-means ++),
  2. přiřad body k nejbližšímu  $c_k$
  3. update  $c_k$ : řešíte přiřazených bodů (popr. obecně minimizace)  
→ reinitializuj, pokud  $c_k$  nemá žády bod.
  4. pokud se něco neměnilo, konec.
- > složitost: za iteraci  $O(Lk)$  počítání vzdálenosti  
 přiřadit  $\underbrace{c_k}_{\text{střed}}$   $\underbrace{\text{cluster}}_{\text{cluster}}$  ~~Když měříme vzdálenost k středu~~
- k-means ++ -> cyklus nahodil první cluster  
 -> další cluster je vybrán z existujících bodů  
 s pravděpodobností jeho vzdálenosti k nejbližšímu středu  
 → pak máme bound na  $E(J) \leq 8(\ln k + 2) J_{OPT}$

k-medians - pro Manhattanovou normu:  $d(a, b) = \sum_{i=1}^n |a_i - b_i|$

-> místo řešit: medians po souřadnicích.

Cukrovka: používáme  $\|x_i - c_k\|_1$ , neumocnit?

-> geometricky medians (je na něj aly).

Bf u.: problem s fátemehf:

$$(c_1, \dots, c_k) = \arg \min_{c_1, \dots, c_k} \sum_{i=1}^n \min_{k=1, \dots, K} d(x_i, c_k).$$

potom shrubování:

$$T_k = \{x \in T : \forall j : d(x, c_k) \leq d(x, c_j)\}.$$

(10)

## EM Algoritmus (Expectation - Maximization).

- základní použití: odhadování směsi Gaussů

- iterativní optimalizace log-likelihood, rozdělena na dva kroky (E a M).

Pro Gaussovské směsi: směs:  $p(x) = \sum_{k=1}^K \pi_k N(x | \mu_k, \Sigma_k)$

1. inicializujeme  $\mu_k, \Sigma_k, \pi_k$

→ pomocí k-means (učebnice shlužky)  
spočítatme v nich hodnoty  $\mu_k, \Sigma_k$   
 $\pi_k$  bude počet bodů v shlužce  
počet bodů)

L "gauss břich" → matice kovarianc  
"responsibility" ažu  $\sqrt{p_{st}}$ , že  $x$   
bylo vygenerováno z Gaussem

2. Expectation step: Počítatme "responsibilities" = posterior  $p_{st}$ , že  $k$ -tý Gauss  
vyprodukoval  $x_i$ . (prostě podělíme  $\pi_k N(x | \mu_k, \Sigma_k)$ )

3. Maximization step: pomocí posteriorních výskytů v E kroku aktualizovat parametry

4. check for convergence, pokud ne, opakovat 2.

## Obeznyj EM:

- matice rozdělení  $p(X, Z | \theta)$ ; X - pozorované proměnné, Z - latenti proměnné  
 $\theta$  - parametry modelu

chceme maximalizovat likelihood  $p(X | \theta)$  podle  $\theta$ .

1. inicializujeme  $\theta_0$

2. E krok: spoaltej:  $p(Z | X, \theta_0)$ .

3. M krok:  $\theta_{n+1} = \operatorname{argmax}_{\theta} \sum_Z p(Z | X, \theta_n) \ln p(X, Z | \theta)$   $\underbrace{\left[ + \ln p(\theta) \right]}$

↑  
tudíme, že toto je dolní odhad maxima  
likelihoodu

4. check convergence.

(77)

73. počítadlo s strojmi,

entropie:  $- \sum_j p_j \log \frac{p_j}{c_j}$

pak se v informacií gain užív.

$$IG = E(P_{\text{unzdroj}}) - \sum_i p_i \cdot E(P_{\text{rozdrojen}})_i$$

(12)

### 3. Odhad pravděpodobnosti: MLE, MAP

parametry

a) zvolíme model (pravděpodobnostní distribuci, např. normální)  $\rightarrow$  počti  $p(x|k)$ .

b) odhadneme jeho parametry  $\hat{\theta} = \arg\max_{\theta} L(\theta) = \arg\max_{\theta} p(T|\theta)$

předpokládáme nezávislost pozorování na parametry pozorování jednotlivých tříd:

$$p(T|\theta) = p(T_1|\theta_1) \cdots p(T_k|\theta_k)$$

$\hookrightarrow x_i | T_i$

Pak analyzujeme pro 1 třídu.

Opět předpokládáme nezávislost tentokrát pozorování:  $\hat{\theta} = \arg\max_{\theta} \prod_{i=1}^N p(x_i|\theta)$ .

(můžeme logaritmovat a dostaneme součet logaritmu).

#### vlastnosti MLE:

- dobré asymptotické vlastnosti: blíží se optimálným parametrem

- při malém počtu pozorování mohou špatně reprezentovat skutečnost.

#### MAP:

- přidáme k MLE prior belief (prior distribution of parameters).

$$\hat{\theta} = \arg\max_{\theta} p(T|\theta)p(\theta)$$

(toto jsme prostě získali z Bayese pro posterior)

- větších chceme, aby prior  $p(\theta)$  měl stejnou funkční podobu jako posterior

- takovýto prioru říkáme conjugate priors  $\rightarrow$  zavedeme v přehledu

- pekné je, že můžeme posterior aktualizovat postupně:

• nové nové měření, použijeme již spočítaný posterior jako prior  
a vypočítáme nový posterior.

#### Bayesovský odhad parametrů:

- minimalizuje risk  $R(\theta)$

- uvažujme TD klasický problém s kvadratickou loss,

$$R(\theta) = \int_{-\infty}^{\infty} p(t|T)(t-\theta)^2 dt$$

$$\hat{\theta}_{MSE} = \arg\min_{\theta} R(\theta) \quad \rightarrow \quad \hat{\theta}_{MSE} = \int_{-\infty}^{\infty} t p(t|T) dt$$