# I13 – Bayesovské rozpoznávání, odhady parametrů metodou maximální věrohodnosti. Učení bez učitele (shlukování, k-means). Regrese. Metoda expectation-maximization. (RPZ)

## Bayesovské rozpoznávání

Založeno na minimalizaci Bayesovského risku R(q).

$$R(q) = \sum_{x \in X} \sum_{k \in K} p_{XK}(x, k) \ W(k, q(x))$$

Bayesovský risk:

$$q^* = \operatorname*{argmin}_{q \in X \to D} R(q)$$

(1)

a **Bayesovská strategie** pak je: množiny: X ... pozorování, D ... rozhodnutí, K ... třídy (skryté stavy)

Shrnutí: Bayesovské rozpoznávání hledá takovou strategii, která minimalizuje ztrátu vyjádřenou Bayesovským riskem. Při 0-1 ztrátové funkci přiřazuje pozorování do stavu s největší aposteriorní pravděpodobností. Nevýhody: Musí existovat smysluplná ztrátová funkce pro daný příklad (jaká je cena za lidský život, apod.), musí být známé apriorní pravděpodobnosti.

- ♦ knows that trams 3, 6, 14, 22, 24 stop at Karlovo namesti
- knows that of those, trams 14 and 24 go to Albertov
- observes the tram type  $x \in \{\text{old}, \text{new}\}$
- knows that the joint probability p(x,k) of a tram of type  $x \in \{\mathsf{old},\mathsf{new}\}$  and line number  $k \in \{3,6,14,22,24\}$  is

	3	6	14	22	24	p(x)
old	0.05	0.15	0.10	0.25	0.05	0.60
new	0.20	0.00	0.05	0.00	0.15	0.40
p(k)	0.25	0.15	0.15	0.25	0.20	

• recall: p(a,b) = p(a|b)p(b) = p(b|a)p(a), where p(a|b) is conditional

#### Average loss (in CZK):

	he decides to run	he decides not to run		
old	112.5	100		
new	75	100		

Thus, the optimal strategy  $q^*(x)$  for each x can obtained as

$$q^*(x) = \underset{d \in D}{\operatorname{argmin}} \sum_{k \in K} p_{Kx}(k \mid x) \ W(k, d).$$

Reject option:

Let us define W(k,d),  $k \in K$ ,  $d \in D$ :

$$W(k,d) = \left\{ \begin{array}{l} 0, & \text{if } d = k \,, \\ 1, & \text{if } d \neq k \text{ and } d \neq \text{not known} \,, \\ \varepsilon, & \text{if } d = \text{not known} \,. \end{array} \right.$$

For R(x, not known), there holds

$$\begin{split} R(x, \text{not known}) &= \sum_{k \in K} p_{K|X}(k \,|\, x) \; W(k, \text{not known}) \\ &= \sum_{k \in K} p_{K|X}(k \,|\, x) \; \varepsilon = \varepsilon \;. \end{split}$$

The decision rule becomes

$$q^*(x) = \begin{cases} \underset{k \in K}{\operatorname{argmax}} \ p_{K|X}(k \,|\, x) \,, & \text{if } 1 - \underset{k \in K}{\max} p_{K|X}(k \,|\, x) < \varepsilon \,, \\ \underset{\text{not known}}{\operatorname{known}} \,, & \text{if } 1 - \underset{k \in K}{\max} p_{K|X}(k \,|\, x) \ge \varepsilon \,. \end{cases}$$

### Strategy $q^*(x)$ can thus be described as follows:

First, find the state k which has the largest a posteriori probability. Let this probability be denoted by  $p_{\text{max}}(x)$ .

If  $1 - p_{\max}(x) < \varepsilon$  then the optimal decision is k.

If  $1-p_{\max}(x) \geq \varepsilon$  then the optimal decision is not known .

### Odhady parametrů metodou maximální věrohodnosti

Vycházíme z předpokladu, že známe "formu" rozložení pravděpodobnosti (např. gaussián, apod.), odhadujeme parametry θ této "formy" (počet parametrů je nízký)

Metoda maximální věrohodnosti (Maximum likelihood estimation = MLE):

$$\hat{\boldsymbol{\theta}} = \operatorname*{argmax}_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{\theta}} p(\mathcal{T}|\boldsymbol{\theta}) \qquad \quad \hat{\boldsymbol{\theta}} = \operatorname*{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^N p(x_i|\boldsymbol{\theta})$$

### Postup matematicky:

- 1. zlogaritmuji (likelihood → log-likelihood): z produktu je suma
  - díky tomuto obvykle následně derivuji součet a ne součin jednodušší
- 2. zderivuji a položím rovno nule

Asymptoticky přechází ke správným parametrům. Odhad parametrů se dělá pro každou třídu zvlášť, postupně.

### **Example 1: Binomial Distribution (ML) (2)**

(copied from the previous slide:)

$$p(R, N|\pi) = \binom{N}{R} \pi^R (1-\pi)^{N-R}.$$

Taking the derivative of  $p(R, N|\pi)$  with respect to  $\pi$  and setting it to zero gives

$$\binom{N}{R} R \pi^{R-1} (1-\pi)^{N-R} - \binom{N}{R} \pi^R (N-R) (1-\pi)^{N-R-1} = 0,$$

and thus

$$R(1-\pi) - (N-R)\pi = 0$$

which implies

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

N = počet pokusů, R = kolikrát chci aby padla 1 na kostce třeba (2x .. )

## Example 2: Normal Distribution (ML) (1)

Let the conditional probability of a class be normal. Assume that the observations  $\mathcal{T}=\{x_1,x_2,...,x_N\}$  are i.i.d. and find the ML estimate for the mean and variance. likelihood to be maximized is:

$$P(\mathcal{T}|\mu,\sigma) = \frac{1}{\sigma^N \sqrt{(2\pi)^N}} \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^N (x_i - \mu)^2\right].$$

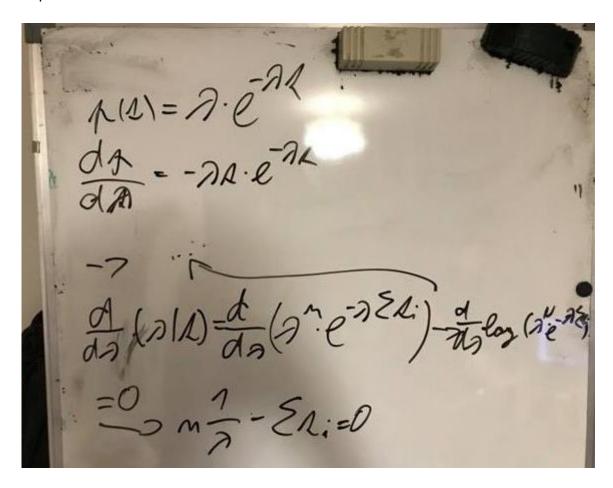
We require that the partial derivatives w.r.t. both  $\mu$  and  $\sigma$  vanish:

$$\frac{\partial P(\mathcal{T}|\mu,\sigma)}{\partial \mu} = P(\mathcal{T}|\mu,\sigma) \frac{1}{\sigma^2} \left( \sum_{i=1}^N (x_i - \mu) \right) = 0$$
$$\frac{\partial P(\mathcal{T}|\mu,\sigma)}{\partial \sigma} = -P(\mathcal{T}|\mu,\sigma) \frac{N}{\sigma} + P(\mathcal{T}|\mu,\sigma) \frac{1}{\sigma^3} \left( \sum_{i=1}^N (x_i - \mu)^2 \right) = 0$$

The first and second equations imply, respectively, the terms on the right. The ML estimator for mean is the sample mean (as before in Example 1) and the ML estimator for variance is the sample variance, with sample mean Eq. (14) plugged into Eq. (15).

$$\begin{split} \hat{\mu}_{\mathrm{ML}} &= \frac{1}{N} \sum_{i=1}^{N} x_i \,, \\ \hat{\sigma}_{\mathrm{ML}}^2 &= \frac{1}{N} \sum_{i=1}^{N} (x_i - \hat{\mu}_{\mathrm{ML}})^2 \,. \end{split}$$

#### Exponenciální rozdělení:



### Učení bez učitele (shlukování, k-means)

Učení bez učitele (unsupervised learning) - pozorování nejsou olabelovaná Semi-supervised - používá se k učení jak olabelovaná, tak neolabelovaná pozorování

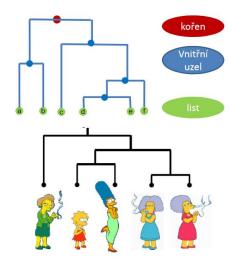
#### Shlukování

Shlukování se používá ke klasifikaci objektů do tříd. Dělí se na hierarchické (vytváří se systém podmnožin, z nichž některé mají další vnitřní členění, tedy podmnožiny nižší hierarchické úrovně) a nehierarchické (množiny jsou disjunktní). Pro příslušnost k daným shlukům je <u>důležitá zvolená metrika</u>, tj. způsob měření vzdálenosti mezi objekty, případně shluky.

Hierarchické metody shlukování zahrnují dva přístupy:

- 1) divizní vychází se z celku, který se člení na menší části
- 2) aglomerativní vychází se z jedinců, spojují se do větších celků

Dendrogram - graficky znázorňuje míru podobnosti v hierarchicky shlukované množině, např.:



Hierarchické shlukování představuje např. metoda K-nejbližších sousedů (K-NN, ta je ale s učitelem), nehierarchická metoda je např. shlukování pomocí algoritmu K-means.

#### K-Means

Data x<sub>i</sub> chceme roztřídit do k tříd, hledáme k center shluků

$$\sum_{k=1}^K \sum_{\mathbf{x} \in \mathcal{T}_k'} \|\mathbf{x} - \mathbf{c}_k'\|^2$$
 (minimalizujeme vzdálenosti od center)

#### Algoritmus:

0. inicializace k centroidů (náhodně/k-means++ (viz. níže))

1. přiřazení bodů k nejbližšímu centroidu:  $\mathcal{T}_k = \{\mathbf{x} \in \mathcal{T} : \forall j, \|\mathbf{x} - \mathbf{c}_k\|^2 \leq \|\mathbf{x} - \mathbf{c}_j\|^2\}$ 

 $\frac{1}{|\mathcal{T}_k|}\sum_{\mathbf{x}\in\mathcal{T}_k}\mathbf{x}\quad\text{if }|\mathcal{T}_k|>0$  2. spočítání nového těžiště shluku:  $\mathbf{c_k}=$ 

3. pokud  $T_k = T_{k+1}$  (stejné clustery jako předchozí iterace), tak konec, jinak goto 1

Lze zobecnit <u>pro jakoukoliv metriku</u> *d* (tedy ne jen normu *I*-2 = Eukleidovskou vzdálenost) Můžeme uvíznout v lokálním extrému (lze řešit více běhy a výběrem nejlepšího řešení)

Algoritmus končí v konečném počtu iterací (je konečný počet stavů = N<sup>K</sup>)

Složitost: I(počet iter)·N(počet dat)·K(počet center)·M(dim dat)

V každé iteraci porovnávám každý bod s každým shlukem → N·K

#### K++

Inicializace pomocí K++ means zvyšuje pravděpodobnost nalezení globálního minima

Algoritmus: c<sub>1</sub> vybrán úplně náhodně,  $d_l = \min_{\mathbf{c} \in \mathcal{C}} \|\mathbf{x}_l - \mathbf{c}\|_{,} p(l) = \frac{d_l^2}{\sum_{l=1}^L d_l^2}_{,}$  dokud nemám K center

### Regrese

**Lineární regrese** je matematická metoda používaná pro proložení souboru bodů v grafu přímkou. O bodech reprezentujících měřená data se předpokládá, že jejich x-ové souřadnice jsou přesné, zatímco

ypsilonové souřadnice mohou být zatíženy náhodnou chybou, přičemž předpokládáme, že závislost y na x lze graficky vyjádřit přímkou.

Lineární regrese představuje aproximaci daných hodnot přímkou metodou nejmenších čtverců. Pokud tuto přímku vyjádříme rovnicí  $y = b_1 + b_2 x$ , jedná se o nalezení optimálních hodnot koeficientů  $b_1$  a  $b_2$ .

V obecnějším případě může lineární regrese znamenat aproximaci daných hodnot (xi, yi) takovou funkcí  $y=f(x,b_1,b_2)$ .

V OPT, přeurčená soustava vedla na regresi, takto: Měli jsme data (měření, známá čísla), z těch bylo nejprve potřeba sestavit matici A. Jednotlivá měření korespondují s řádky matice A. Maticí A budu násobit vektor neznámých parametrů, které se snažím odhadovat. Na pravé straně rovnice je vektor výsledků měření y. Matlab vyřeší soustavu pomocí příkazu: θ = A\y (y je vektor "výsledků" měření)

Interpolace polynomem:

$$f(x, \boldsymbol{\theta}) = \theta_1 + \theta_2 x + \theta_3 x^2 + \dots + \theta_n x^{n-1}$$

$$\mathbf{A} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{n-1} \\ & & \vdots & & & \\ 1 & x_m & x_m^2 & \dots & x_m^{n-1} \end{bmatrix}$$

### Logistická regrese – Z RPZ:

Lineární klasifikátor, log odds:

$$q(\mathbf{x}) = 1$$
 if  $p(1|\mathbf{x}) > p(2|\mathbf{x})$   $\Leftrightarrow$   $\frac{p(1|\mathbf{x})}{p(2|\mathbf{x})} > 1$   $\Leftrightarrow$   $\ln \frac{p(1|\mathbf{x})}{p(2|\mathbf{x})} > 0$ 

a obrácené znaménko pro q(x) = 2

Pokud je poměr lineární, přecházíme na problém hledání parametrů dělicí nadroviny:  $w=[w_0,w], x=[1,x]$ 

$$\mathbf{w}^{\mathsf{T}}\mathbf{x} > 0 \rightarrow \mathbf{q}(\mathbf{x}) = 1$$

Prior probabilities způsobí jen posun (jsou obsaženy v členu w<sub>0</sub>)

Log odds: 
$$\ln \frac{p(1|\mathbf{x})}{p(2|\mathbf{x})} = \mathbf{w} \cdot \mathbf{x}$$

$$\rightarrow p(1|\mathbf{x}) = p(2|\mathbf{x}) \exp(\mathbf{w} \cdot \mathbf{x}), \ p(2|\mathbf{x}) = p(1|\mathbf{x}) \exp(-\mathbf{w} \cdot \mathbf{x})$$

$$\rightarrow 1 = p(1|\mathbf{x}) + p(2|\mathbf{x}) = p(1|\mathbf{x}) \ (1 + \exp(-\mathbf{w} \cdot \mathbf{x})) = p(2|\mathbf{x}) \ (1 + \exp(\mathbf{w} \cdot \mathbf{x}))$$

$$k : \{-1,1\} \rightarrow p(k|\mathbf{x}) = \frac{1}{1 + e^{-k\mathbf{w} \cdot \mathbf{x}}}$$
 (sigmoida)

#### Algoritmus:

+

0. upravit trénovací množinu:  $\{(x_i, y_i)\}$ , x z  $\mathbb{R}^D$ , y = +-1, na  $\{y_i^*(x_i, 1)\}$ . Init w z  $\mathbb{R}^{D+1}$ 

1. minimalizujeme funkci  $E(w) = -I(w) = Sum_i \ln(1 + e^{-wx})$ Vypočítáme gradient  $\nabla_w E = -Sum_i x^* e^{-wx}/(1 + e^{-wx})$ 

Updatujeme w:= w - μ\*∇<sub>w</sub>E (jdeme proti směru grad, protože minimalizujeme E)

Pokud  $E(w_{nov\acute{e}}) < E(w_{star\acute{e}})$ : step = 2\*step, jinak: step = step/2 a neupdatuj w,g,E 2. goto 1

Diskriminační funkce pro 2 třídy se stejnou kovarianční maticí

$$\begin{split} \ln \frac{p(1|\mathbf{x})}{p(2|\mathbf{x})} &= \ln \frac{p(\mathbf{x}|1)}{p(\mathbf{x}|2)} \frac{p(1)}{p(2)} = \ln \frac{p(\mathbf{x}|1)}{p(\mathbf{x}|2)} + \ln \frac{p(1)}{p(2)} \\ & \ln \frac{p(1|\mathbf{x})}{p(2|\mathbf{x})} = \ln \frac{(2\pi)^{-\frac{D}{2}} |\mathbf{\Sigma}|^{-\frac{1}{2}}}{(2\pi)^{-\frac{D}{2}} |\mathbf{\Sigma}|^{-\frac{1}{2}}} e^{-\frac{1}{2} \left\{ (\mathbf{x} - \boldsymbol{\mu}_1)^\top \mathbf{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_1) - (\mathbf{x} - \boldsymbol{\mu}_2)^\top \mathbf{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}_2) \right\}} + \ln \frac{p(1)}{p(2)} \\ & \circ \quad \ln(e^{\mathbf{x}}) = \mathbf{x} \; ; \; \text{maticov\'e rozn\'asobit} \; ; \; \mathbf{x}^T \mathbf{\Sigma}^{-1} \mathbf{x} \; \text{se po\'erou} \; ; \; \mathbf{x}^T \mathbf{\Sigma}^{-1} \mathbf{\mu} = \boldsymbol{\mu}^T \mathbf{\Sigma}^{-1} \mathbf{x} \; \text{(proto\'ee } \mathbf{\Sigma} \; \text{je symetrick\'a} : } \mathbf{\Sigma} = \mathbf{\Sigma}^T) \\ & \underbrace{\left[ (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \mathbf{\Sigma}^{-1} \right] \mathbf{x} + \underbrace{\frac{1}{2} (\boldsymbol{\mu}_1^\top \mathbf{\Sigma}^{-1} \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2^\top \mathbf{\Sigma}^{-1} \boldsymbol{\mu}_2) + \ln \frac{p(1)}{p(2)}}_{\mathbf{W}} \right]}_{\mathbf{W}} \\ & \to \mathbf{W} \cdot \mathbf{x} + \mathbf{w}_0 \; , \; \text{kde} \; \mathbf{w} \in \mathbf{R}^{\mathbf{D}}, \; \mathbf{w}_0 \in \mathbf{R} \end{split}$$

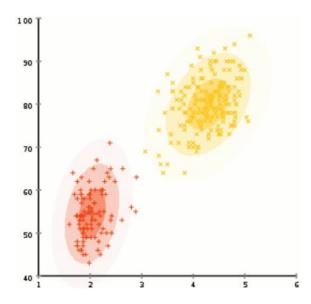
Závěr: Pro 2 třídy s norm. rozdělením se stejnými kovariančními maticemi: log odds je lineární funkce Kovarianční matice: → vl. čísla (a vektory): vidím osy elipsy Gausse, velké vl. č. = velká var v daném směru

## **Metoda expectation-maximization**

EM je iterační metoda pro hledání MLE nebo MAP odhadu parametrů. (MLE = maximálně věrohodný odhad, MAP = maximální aposteriorní pravděpodobnost).

Při EM iteracích se pravidelně střídají kroky výpočtu střední hodnoty (očekávání, E) s kroky maximalizace (M). V kroku E se vytváří očekávaná logaritmická věrohodnostní funkce na základě aktuálního odhadu parametrů. V kroku M se počítají parametry maximalizující očekávanou logaritmickou věrohodnostní funkci nalezenou v kroku E.

Odhady parametrů se používají pro určení rozdělení skrytých proměnných (latent variables).



EM se často používá pro shlukovou analýzu ve strojovém učení a počítačovém vidění. Použití:

- Missing data měření jsou nekompletní (pro některé x chybí nějaká složka)
- Latent variables měření kompletní, ale bylo by mnohem lepší kdyby byla ještě nějaká informace, např. mix více rozdělení typicky

Jako k-means (nemáme informaci o třídě).

Častá aplikace: odhad mixu Gaussů

Dále, viz. I12 – EM Algorithm.

#### Algoritmus (S2: "E-step", S3: "M-step")

- 1. Initialize  $\mu_c$ ,  $\mu_t$
- 2. Compute  $q(z_i) = p(z_i|y_i, \mu_c, \mu_t)$  for all i = 1, 2, ..., M
- 3. Recompute  $\mu_c$ ,  $\mu_t$  according to Eqs.(41, 42)
- 4. If termination condition is met, finish. Otherwise goto 2. ( $\mu_c$ ,  $\mu_t$  viz. níže (2))  $q^{(t+1)} = \operatorname*{argmax}_{q} \mathcal{L}(q, \boldsymbol{\theta}^{(t)}), \quad \boldsymbol{\theta}^{(t+1)} = \operatorname*{argmax}_{\boldsymbol{\theta}} \mathcal{L}(q^{(t+1)}, \boldsymbol{\theta})$

E-step počítá pravděpodobnost, že daný bod je generovaný danými komponenty (jeho aposteriorní pravděpodobnost, že patří k dané třídě)

**M-step** počítá neznámé parametry analyticky (odhad parametrů pomocí MLE/MAP) je užitečný a efektivní dále pro exponenciální rozdělení chytře maximalizuje likelihood tím, že tlačí lower bound nahoru

iterativní metoda ⇒ <u>nemusí skončit v globálním maximu</u> opatrně inicializovat (stejně jako u K-means, NNs, ...)

#### Příklad car/truck:

- 0. init  $\mu_c$ ,  $\mu_t$

$$\mu_{\rm c} = \frac{\sum_{i=1}^M q(z_i={\rm car})\,y_i^\bullet}{\sum_{i=1}^M q(z_i={\rm car})}$$
 • 2.  $\mu_{\rm c}$ ,  $\mu_{\rm t}$ :

- ukončit pokud je termination condition splněna
- k-means se liší pouze v bodě 1 tam se přiřazuje pouze 1 nebo 0

$$q(z_i = \mathsf{car}) = \llbracket |y_i^{ullet} - \mu_\mathsf{c}| < |y_i^{ullet} - \mu_\mathsf{t}| 
rbracket$$