

Interakce člověk–počítač v přirozeném jazyce (ICP)

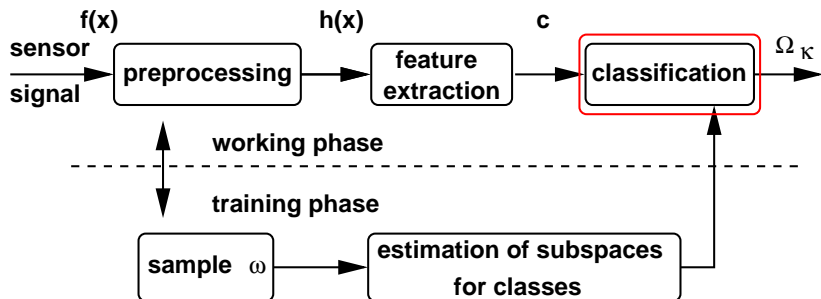
LS 2013 — Klasifikace

Tino Haderlein, Elmar Nöth

Katedra informatiky a výpočetní techniky (KIV)
Západočeská univerzita v Plzni

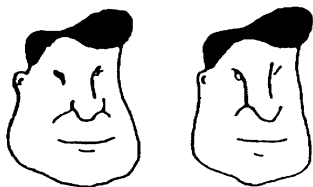
Lehrstuhl für Mustererkennung (LME)
Friedrich-Alexander-Universität Erlangen-Nürnberg

Klasifikační systém



Similar Patterns

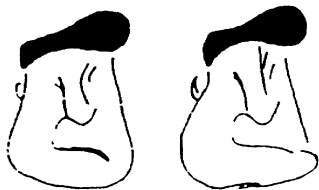
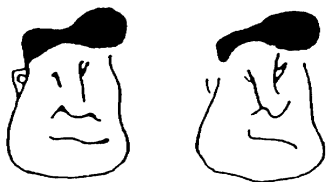
Postulate 4:
two patterns are similar, if an appropriately defined
distance measure between their feature vectors is small.



Similar Patterns

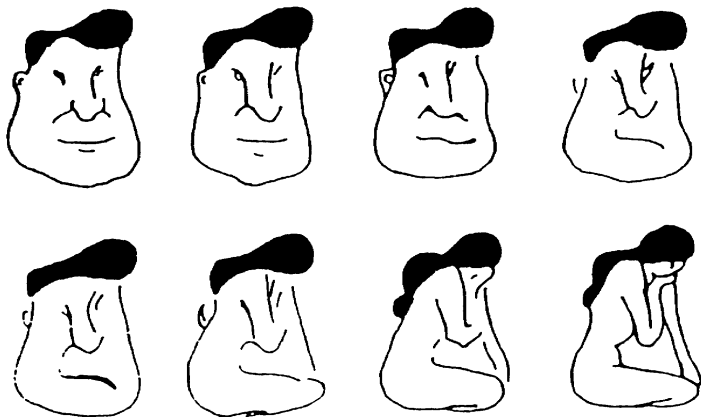
Postulate 4:

two patterns are similar, if an appropriately defined distance measure between their feature vectors is small.



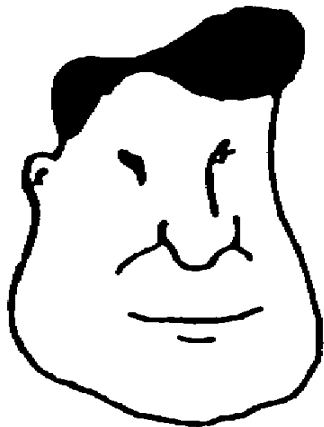
Similar Patterns

Postulate 4:
two patterns are similar, if an appropriately defined
distance measure between their feature vectors is small.



Similar Patterns

Proof that old men look like young women



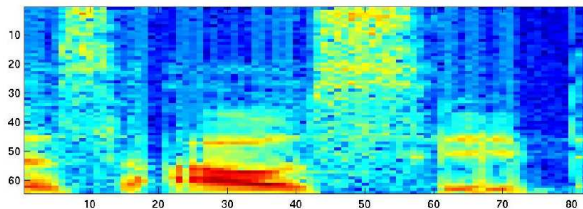
Míry podobnosti – vzdálenosti

- Promluva, která má být klasifikována do jedné třídy Ω_{κ} je teď posloupnost pozorování s $\mathbf{c} \in \mathbb{R}^n$, $n \approx 20$.
- Klasifikace znamená, že každé pozorování \mathbf{c} musí být srovnáno s vhodným referenčním vektorem každé možné třídy Ω_{κ} .
- Klasifikátor musí počítat průměrnou podobnost přes všechna pozorování, která patří k promluvě.
- Když vhodný referenční vektor $\mathbf{c}_{\Omega_{\kappa}}$ třídy Ω_{κ} je považován za bod v \mathbb{R}^n , pak míra podobnosti mezi $\mathbf{c}_{\Omega_{\kappa}}$ a aktuálním pozorovacím vektorem \mathbf{c}_j může být jejich vzdálenost podle nějaké metriky:
 - $d_{\kappa j} = \sum_{i=1}^n |\mathbf{c}_{\Omega_{\kappa}}(i) - \mathbf{c}_j(i)|$ cityblock metric
 - $d_{\kappa j} = \sqrt{\sum_{i=1}^n (\mathbf{c}_{\Omega_{\kappa}}(i) - \mathbf{c}_j(i))^2}$ euklidovská vzdálenost
- Rozhodni se pro κ s minimální vzdáleností, tj. nejvyšší podobností.

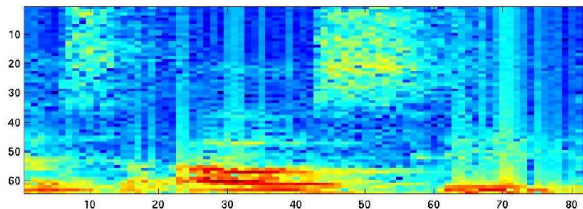
Míry podobnosti – hustoty pravděpodobnosti

- považuj vhodný referenční vektor $\mathbf{c}_{\Omega_{\kappa}}$ třídy κ za stochastický proces, tj. podmíněná hustota pravděpodobnosti $P(\mathbf{c}|\Omega_{\kappa})$ může být popsána rodinou parametrických funkcí
- proč stochastický proces?
- určité faktory, které ovlivňují příznakový vektor, jsou neznámé a/nebo není možné je modelovat explicitně
- např. klasifikační úloha: klasifikace mluvčího
- příklady ovlivňujících faktorů:
 - emocionální stav
 - stav uživatele (user state)
 - mikrofon 1
 - mikrofon 2

Stejná promluva, různé mikrofony

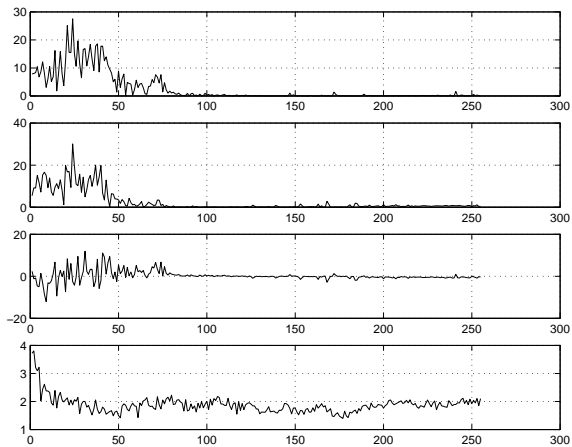


mikrofon
blízko
mluvčího
(close talk
microphone)



prostorový
mikrofon
(room
microphone)

/a/, different microphones



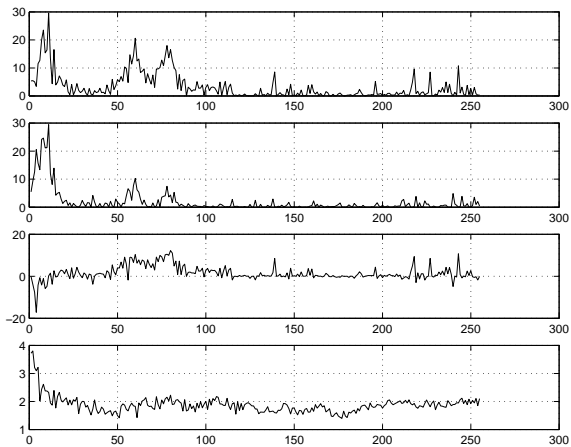
close talk
microphone

omni-directional
microphone

difference

average
difference
(10 sec)

/i/, different microphones



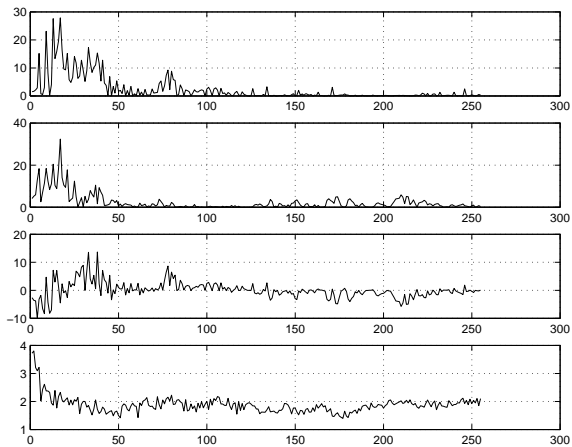
close talk
microphone

omni-directional
microphone

difference

average
difference
(10 sec)

/u/, different microphones



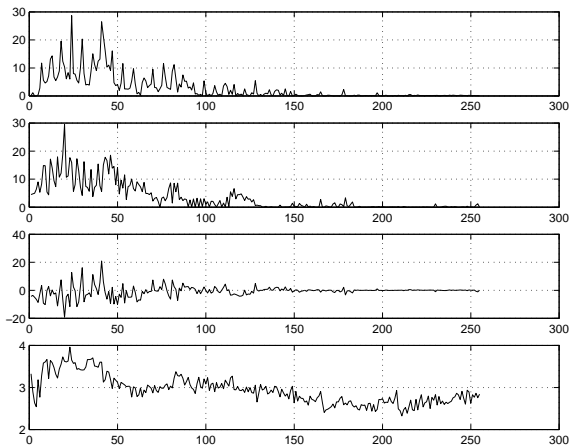
close talk
microphone

omni-directional
microphone

difference

average
difference
(10 sec)

/a/, close-talk microphone, different speakers



speaker
A

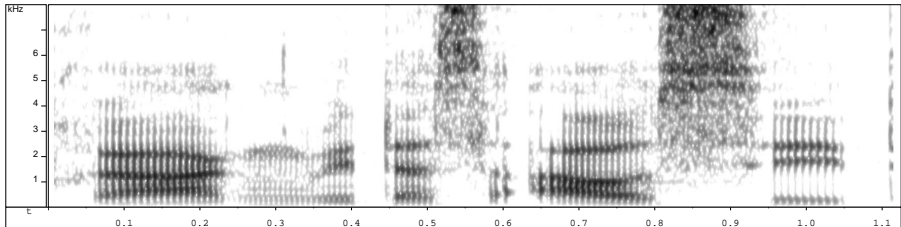
speaker
B

difference

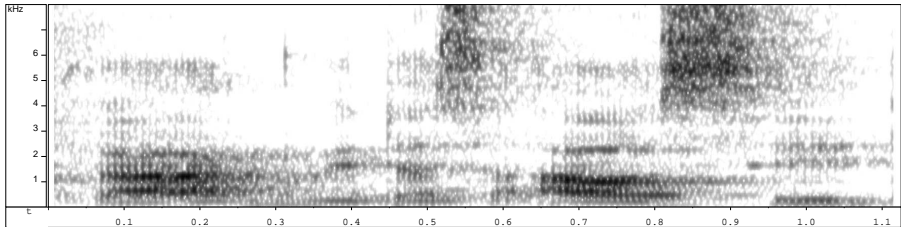
average
difference
(10 sec)

Close-talk vs. room microphone

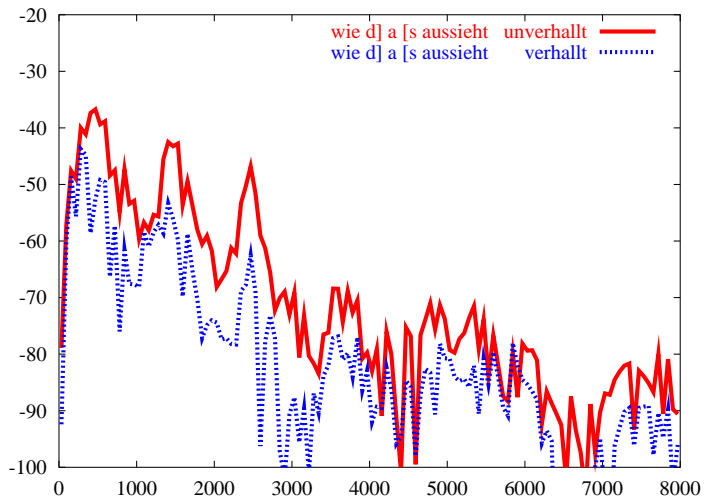
... fragen wie das aussieht ... Nahbesprechungsmikrofon, sp1CTalk-006, Muedigkeitsstichprobe



... fragen wie das aussieht ... Raummikrofon, sp1Mic07-006, Muedigkeitsstichprobe

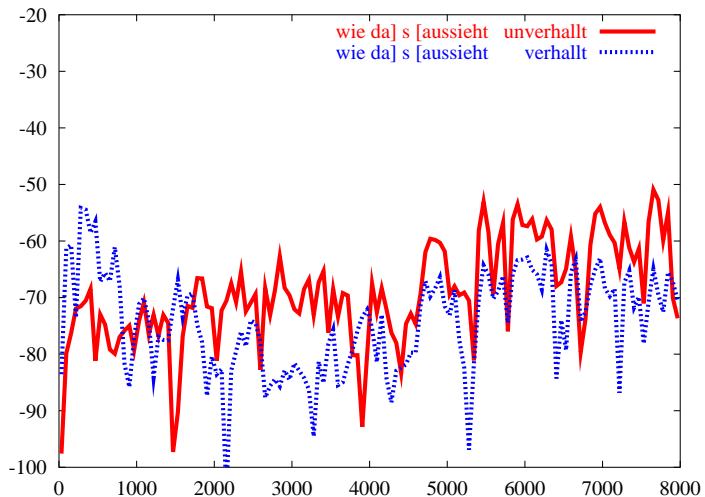


Close-talk vs. room microphone



spektrogram hlásky ‚a‘ z promluvy „wie das aussieht“; unverhált=bez dozvuku, verhált=s dozvkem

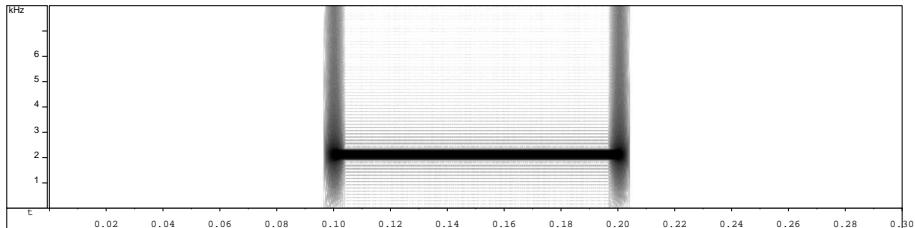
Close-talk vs. room microphone



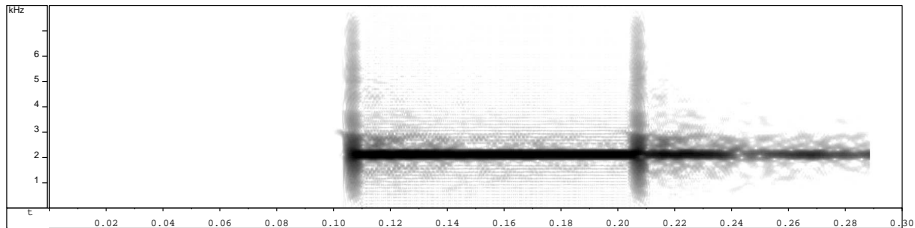
spektrogram hlásky ‚s‘ z promluvy „wie das aussieht“; unverhallt=bez dozvuku, verhallt=s dozvukem

Close-talk vs. room microphone

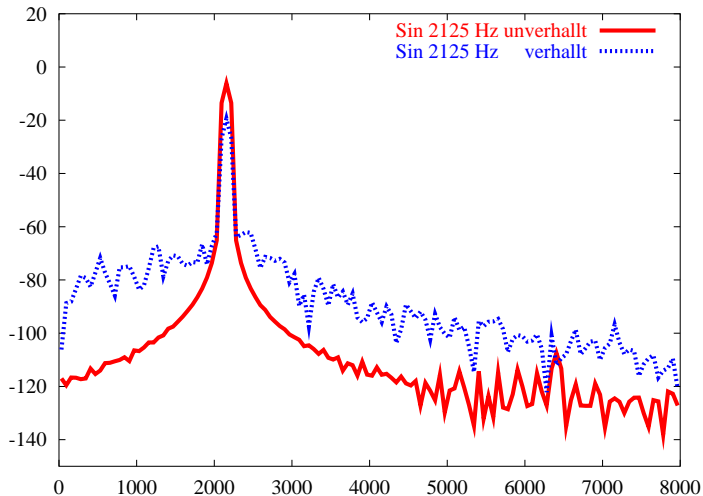
File: sin2125.wav Page: 1 of 1 Printed: Fri Dec 26 00:18:39



File: sin2125-verh.wav Page: 1 of 1 Printed: Fri Dec 26 00:18:07

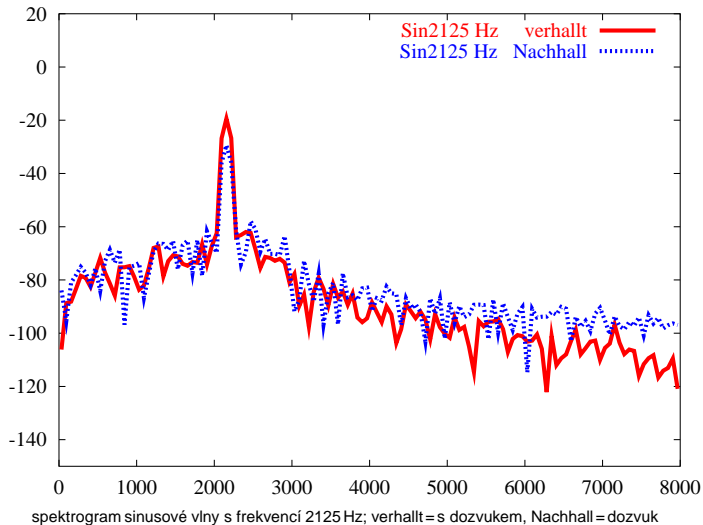


Close-talk vs. room microphone



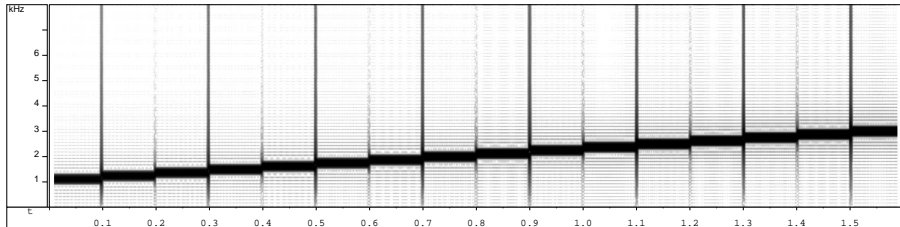
spektrogram sinusové vlny s frekvencí 2125 Hz; unverhällt= bez dozvuku, verhällt= s dozvukem

Close-talk vs. room microphone

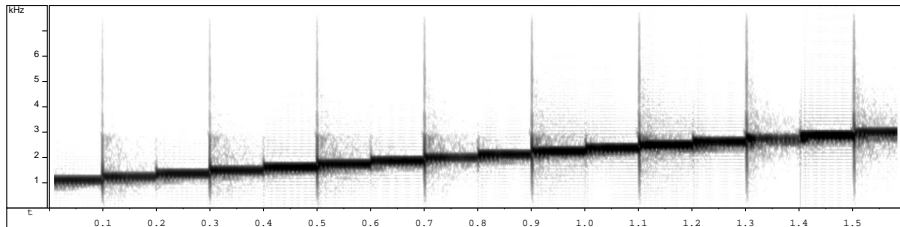


Close-talk vs. room microphone

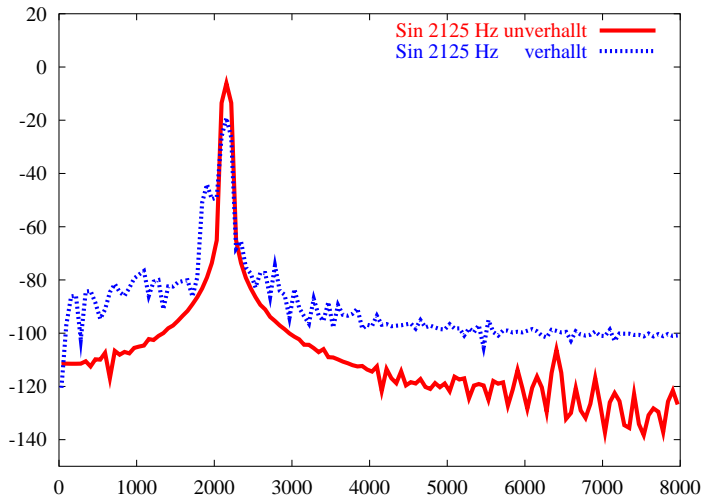
File: Page: 1 of 1 Printed: Fri Dec 26 23:29:02



File: Page: 1 of 1 Printed: Fri Dec 26 23:34:51



Close-talk vs. room microphone



spektrogram sinusové vlny 2125 Hz z posloupnosti stoupajících frekvencí; unverhált=bez dozvuku, verhált=s dozvukem

Míry podobnosti – hustoty pravděpodobnosti

- Buď prototypická realizace daného zvuku od daného mluvčího, který je mírně obměněn ovlivňujícími faktory, které jsou neznámé a jsou proto považované za náhodný proces, který má jako základ tuto prototypickou realizaci.
- Chování náhodného procesu je popsáno hustotou pravděpodobnosti (probability density function; PDF).
- Buď rodina parametrických funkcí.
- Parametry se trénuje representační množinou ω .
- Nejčastěji užívaná rodina parametrických funkcí je rodina Gaußových* funkcí (normální rozdělení), charakterizované průměrným vektorem μ a kovarianční maticí \mathbf{K} .

*: německy „Gauß“ s tzv. „ostrým s“, mezinárodně „Gauss“

Tréninková množina

- dva mluvčí sp_i
- jazyk s třemi hláskami sou_j
- Každá hláska je pro trénování každým mluvčím řečena pětkrát.
- Pro každý tréninkový vzor: Mluvčí je znám, ale hláska ne.
- Každý vzor je popsán jako dvojrozměrný příznakový vektor $\mathbf{c}_i \in \mathbb{R}^2$.
- Chceme popisovat chování náhodné proměnné \mathbf{c} dvojdimenzionální Gaußovou funkcí:

$$P(\mathbf{c}|sp_i, sou_j) = \mathcal{N}(\mathbf{c}|\mu_{ij}, \mathbf{K}_{ij}) = \sqrt{|2\pi\mathbf{K}_{ij}|} \exp\left[-\frac{(\mathbf{c}-\mu_{ij})_t \mathbf{K}_{ij}^{-1} (\mathbf{c}-\mu_{ij})}{2}\right]$$

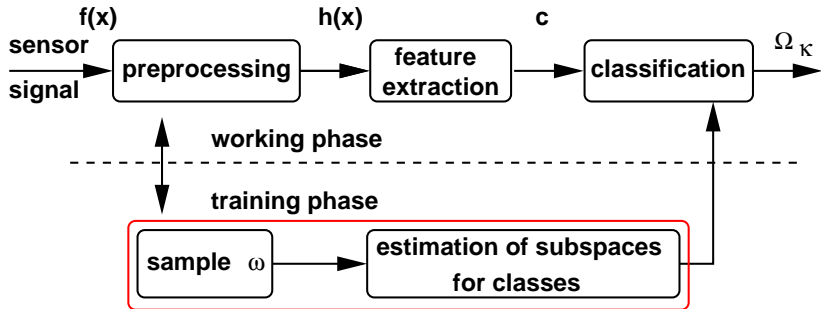
- Problém: Aktuální hláska není známa.
- Řešení: Užívej **jednu** pravděpodobnostní hustotní funkci, která je lineární kombinací Gaußových funkcí:

Gaußův model nebo Gaussian Mixture Model (GMM)

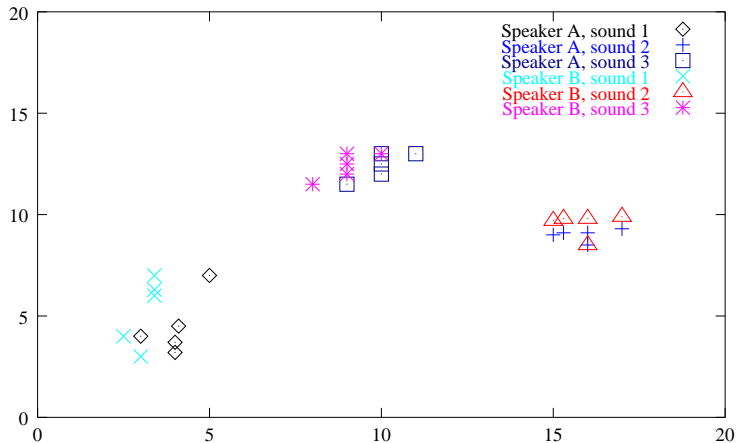
$$P(\mathbf{c}|sp_i) = \sum_{k=1}^{K_i} \omega_{ik} \cdot \mathcal{N}(\mathbf{c}|\mu_{ik}, \mathbf{K}_{ik})$$

- μ , \mathbf{K} a váhy ω se mohou trénovat účinně (algoritmem EM).

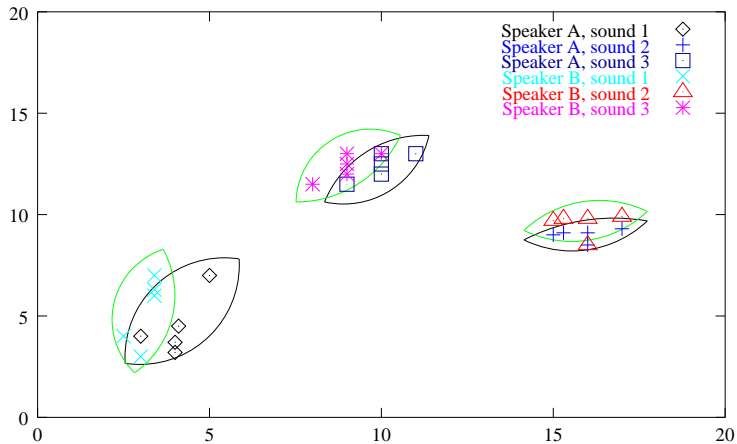
Klasifikační systém



Estimates for PDFs based on a Training Sample

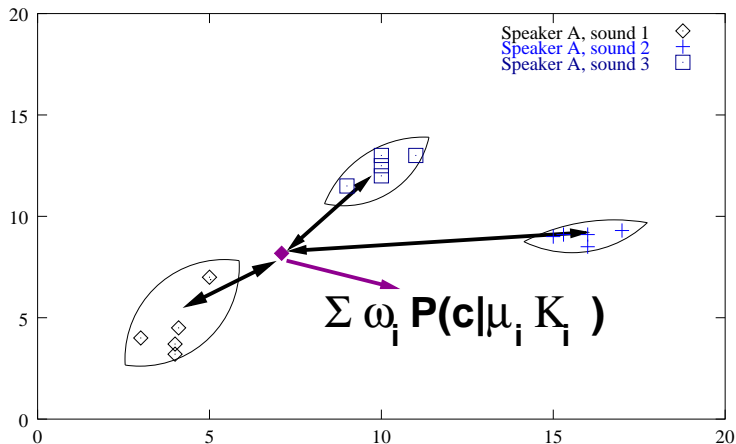


Training Sample



Classification of One Frame w.r.t. Speaker A

Result is an estimate for the conditional probability
 $P(\mathbf{c}|\text{speakerA})$



Základní technologie klasifikace

- Klasifikuj každé okno (frame) promluvy vzhledem ke všem třídám (mluvčí).
- Pro každou třídu zkombinuj klasifikační výsledek pro každý frame (suma, násobení, ...).
- Když se užívá míru vzdálenosti, rozříd' třídy vzestupně (začni s minimální vzdáleností).
- Když se užívá podmíněnou funkci pravděpodobnosti, rozříd' třídy sestupně (začni s maximální pravděpodobností).
- Výsledek je tříděný seznam tříd s hodnoceními (scores).

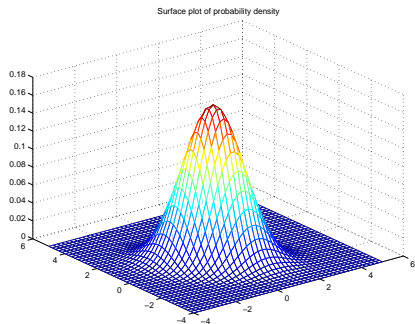
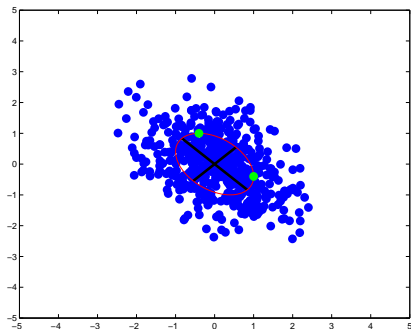
Gaußovo rozdělení

- nejčastěji užívaný předpoklad rozdělení příznakových vektorů
- definice *Gaußova* nebo *normálního rozdělení*

$$P(\mathbf{x} \mid \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}_m|}} \cdot e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_m)^\top \boldsymbol{\Sigma}_m^{-1}(\mathbf{x}-\boldsymbol{\mu}_m)}$$

- průměrný vektor $\boldsymbol{\mu}_m$
- symetrická, pozitivně semidefinitní kovarianční matice $\boldsymbol{\Sigma}_m$

Gaußovo rozdělení



Separáčn funkce Gauova klasifikátoru

přklad:

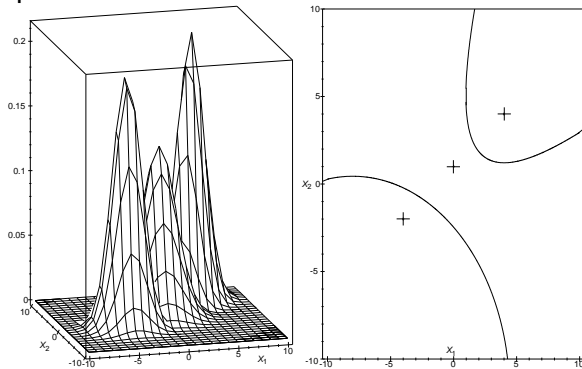
$$\mu_1 = \begin{pmatrix} -4 \\ -2 \end{pmatrix}, \Sigma_1 = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix},$$

$$\mu_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \Sigma_2 = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix},$$

$$\mu_3 = \begin{pmatrix} 4 \\ 4 \end{pmatrix}, \Sigma_3 = \begin{bmatrix} 2,5 & 1,5 \\ 1,5 & 2 \end{bmatrix}.$$

Separační funkce Gaußova klasifikátoru

Příklad: Hustotní funkce tří předem definovaných Gaußových rozdělí (vlevo) a separační čáry mezi rozdělí (vpravo; křížky = průměry $\mu_i, i \in \{1, 2, 3\}$), když všechny tři třídy mají stejnou *a priori* pravděpodobnost:



Identifikace vážené sumy rozdělení

Předpoklad:

Vektory \mathbf{x} , o kterých není známo, ke které třídě κ patří, jsou rozdělené podle **marginální hustoty**:

$$P(\mathbf{x} \mid \mathbf{p}, \theta) = \sum_{\kappa=1}^K p_{\kappa} \cdot P(\mathbf{x} \mid \theta_{\kappa})$$

Jsou \mathbf{p} , θ jednoznačně určitelné, když je postup funkce $P(\mathbf{x} \mid \mathbf{p}, \theta)$ znám?

Ano – když rodina $P(\mathbf{x} \mid \theta)$ tvoří bázi odpovídajícího prostoru funkcí; pak existuje jednoznačná lineární kombinace $P(\mathbf{x})$.

Rodina $\{\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})\}$ multivariátních normálních hustot je **ortogonální bází**.

Neexistuje však algoritmus na výpočet báze.

Gaußovo rozdělení

- také nazýváno jako GMM = Gaussian Mixture Model
- několik Gaußových hustot je vážené a shrnuty
- také multimodální hustoty můžeme tak reprezentovat
- definice hustoty *GMM*

$$P(\mathbf{x} \mid p(1), \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1, \dots, p(M), \boldsymbol{\mu}_M, \boldsymbol{\Sigma}_M) = \sum_{m=1}^M p(m) \cdot \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)$$

- $p(m)$: a priori pravděpodobnosti jednotlivých hustot
- platí: $\sum_{m=1}^M p(m) = 1$
- problém: stanovit počet M normálních rozdělení
- maximum likelihood odhad parametrů pomocí algoritmu EM

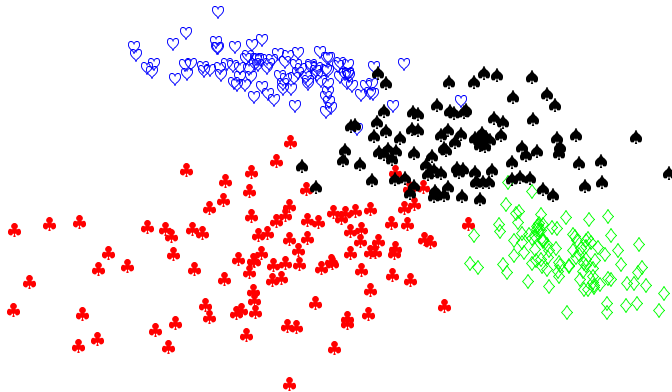
Algoritmus EM

pozorování v \mathbb{R}^2



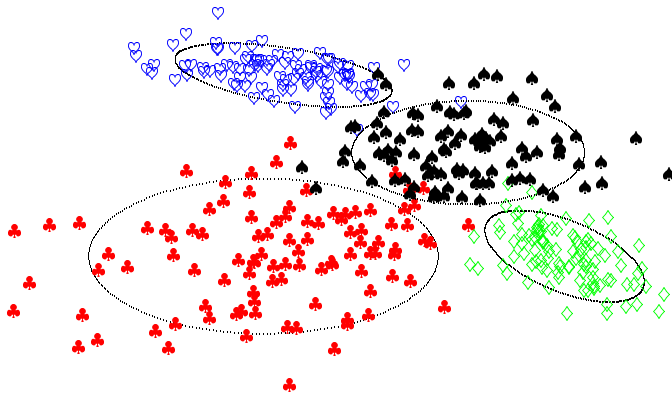
Algoritmus EM

náhodně tvořené vektory Gaußovy sumy 4 hustot v \mathbb{R}^2



Algoritmus EM

μ, σ počítané pomocí EM



Odhad průměrových vektorů a rozptylových matic

odhad maximální věrohodnosti (maximum likelihood, ML) podle vzorů

Maximalizuj logaritmovanou pravděpodobnost pro produkci všech prvků tréninkového vzorku $\omega = (\mathbf{y}_1, \dots, \mathbf{y}_N)$ s vedlejší podmínkou $\sum_{\kappa} p_{\kappa} = 1$

$$\begin{aligned}\mathcal{L}_{\text{MIX}}(\mathbf{p}, \boldsymbol{\theta}) &= \log P(\omega \mid \mathbf{p}, \boldsymbol{\theta}) \\ &= \log \prod_{i=1}^N P(\mathbf{y}_i \mid \mathbf{p}, \boldsymbol{\theta}) \\ &= \sum_{i=1}^N \log \left(\sum_{\kappa=1}^K p_{\kappa} P(\mathbf{y}_i \mid \boldsymbol{\theta}_{\kappa}) \right) .\end{aligned}$$

Odhad průměrových vektorů a rozptylových matic

⇒ systém **sdužených, transcendentních rovnic** pro $\hat{\rho}_\kappa$, $\hat{\mu}_\kappa$ a $\hat{\Sigma}_\kappa$
 (předpoklad: normální rozdělení

$$\text{např. odhad pro } \hat{\rho}_\kappa = \frac{1}{N} \sum_{i=1}^N P(\Omega_\kappa | \mathbf{y}_i, \hat{\boldsymbol{\rho}}, \hat{\boldsymbol{\theta}})$$

označený (labeled) tréninkový vzorek

⇒ $P(\Omega_\kappa | \mathbf{y}_i, \hat{\boldsymbol{\rho}}, \hat{\boldsymbol{\theta}}) = 1$ pro $\mathbf{y}_i \in \Omega_\kappa$, 0 jinak

úkol při „učení bez učitele“:

náhodný proces tvoří dvojice hodnot (κ, \mathbf{x}) náhodných proměnných (Ω, \mathbf{X}) s hustotou pravděpodobnosti $P(\kappa, \mathbf{x})$

Problém: Hodnoty diskretních náhodných proměnných Ω **nejsou pozorovatelné!**

Algoritmus EM

máme:

náhodný proces (\mathbf{X}, \mathbf{U}) s hustotou rozdělení $P(\mathbf{x}, \mathbf{u} | \theta)$
(\mathbf{x} pozorovatelný a \mathbf{u} skrytý)

hledáme:

ML odhad neznámých parametrů rozdělení θ
pomocí maximalizace veličiny

$$\mathcal{L}_{EM}(\theta) = \log P(\mathbf{x} | \theta) = \log \int_{\mathbf{u}} P(\mathbf{x}, \mathbf{u} | \theta) d\mathbf{u}$$

Algoritmus EM

chicken-egg problem:

$\theta \leftrightarrow \mathbf{u} \leftrightarrow \theta \leftrightarrow \mathbf{u} \leftrightarrow \theta \leftrightarrow \mathbf{u} \leftrightarrow \dots$

optimalizace veličiny θ vyžaduje hodnotu nebo rozdělení skryté náhodné proměnné, rozdělení procesu \mathbf{U} je závislé na této θ

iterativní východisko:

algoritmus EM (expectation maximization)

účel:

postupné zlepšení parametrů vzhledem k cílové funkci algoritmu ML

Algoritmus EM

platí vztah

$$\begin{aligned}P(\mathbf{x}, \mathbf{u} | \hat{\theta}) &= P(\mathbf{u} | \mathbf{x}, \hat{\theta})P(\mathbf{x} | \hat{\theta}) \implies \\ \log P(\mathbf{x}, \mathbf{u} | \hat{\theta}) &= \log P(\mathbf{u} | \mathbf{x}, \hat{\theta}) + \log P(\mathbf{x} | \hat{\theta}) \implies \\ \log P(\mathbf{x} | \hat{\theta}) &= \log P(\mathbf{x}, \mathbf{u} | \hat{\theta}) - \log P(\mathbf{u} | \mathbf{x}, \hat{\theta})\end{aligned}$$

definice **Kullback-Leiblerovy difference** $Q(\theta, \hat{\theta})$,
podmíněné očekávání všech dat (\mathbf{x}, \mathbf{u})

Algorithmus EM

$$\begin{aligned}\mathcal{L}_{\text{EM}}(\hat{\theta}) &= \log P(\mathbf{x} | \hat{\theta}) \\ &= \int_{\mathbf{u}} \log P(\mathbf{x} | \hat{\theta}) P(\mathbf{u} | \mathbf{x}, \theta) d\mathbf{u} \\ &= E[\log P(\mathbf{x} | \hat{\theta}) | \mathbf{x}, \theta] \\ &= E[\log P(\mathbf{x}, \mathbf{u} | \hat{\theta}) - \log P(\mathbf{u} | \mathbf{x}, \hat{\theta}) | \mathbf{x}, \theta] \\ &= \underbrace{E[\log P(\mathbf{x}, \mathbf{u} | \hat{\theta}) | \mathbf{x}, \theta]}_{=: Q(\theta, \hat{\theta})} - \underbrace{E[\log P(\mathbf{u} | \mathbf{x}, \hat{\theta}) | \mathbf{x}, \theta]}_{=: H(\theta, \hat{\theta})}\end{aligned}$$

Algoritmus EM

platí: $\log q \leq q - 1$

⇒ **Jensenova nerovnost** pro podmíněnou entropii $H(\theta, \hat{\theta})$:

$$\begin{aligned}
 H(\theta, \hat{\theta}) - H(\theta, \theta) &= \int_{\mathbf{u}} \left(\log \frac{P(\mathbf{u} | \mathbf{x}, \hat{\theta})}{P(\mathbf{u} | \mathbf{x}, \theta)} \right) P(\mathbf{u} | \mathbf{x}, \theta) d\mathbf{u} \\
 &\leq \int_{\mathbf{u}} \left(\frac{P(\mathbf{u} | \mathbf{x}, \hat{\theta})}{P(\mathbf{u} | \mathbf{x}, \theta)} - 1 \right) P(\mathbf{u} | \mathbf{x}, \theta) d\mathbf{u} \\
 &= \int_{\mathbf{u}} P(\mathbf{u} | \mathbf{x}, \hat{\theta}) d\mathbf{u} - \int_{\mathbf{u}} P(\mathbf{u} | \mathbf{x}, \theta) d\mathbf{u} \\
 &= 1 - 1 \\
 &= 0.
 \end{aligned}$$

Algoritmus EM

$$\Rightarrow H(\theta, \hat{\theta}) \leq H(\theta, \theta)$$

\Rightarrow **dostatečné kritérium pro iterativní zlepšení parametrů:**

$$Q(\theta, \hat{\theta}) \geq Q(\theta, \theta) \quad \Rightarrow \quad \mathcal{L}_{EM}(\hat{\theta}) \geq \mathcal{L}_{EM}(\theta)$$

vylepšená odhadní hodnota ML: θ^* s podmínkou $Q(\theta, \cdot) \rightarrow \max$

Algoritmus EM

1 Stanov vhodné počáteční parametry \hat{B}^0 .

2 Pro $i = 1, 2, 3, \dots$:

(1) Krok expectation: Stanov rozdělení $\gamma^i(Y)$ a posteriori pravděpodobnosti skryté náhodné proměnné Y :

$$\gamma^i(Y) = P(Y | X, \hat{B}^{i-1})$$

(2) Krok maximization: Maximalizuj Kullback-Leiblerovu diferenci Q :

$$\hat{B}^i = \operatorname{argmax}_{\hat{B}} \int_Y \gamma^i(Y) \cdot \log P(X, Y | \hat{B}) dY$$

(3) Kontroluj vhodnou terminační podmínku.

Algoritmus EM pro GMM

- Problém při odhadu ML parametrů GMM: Přiřazení příznakového vektoru \mathbf{x} k jedné z M hustot není známo.
- Algoritmus EM řeší problém iterativní optimalizací.
- Není garantováno, že tak najdeme *nejlepší* parametry, které maximalizují pravděpodobnost tréninkového vzorku.
- Ale nacházíme vždycky *lokální* optimum, tj. parametry, které jsou aspoň tak dobré jako iniciální hodnoty.
- Zkušenost: Nacházíme velmi dobré parametry, když jsou iniciální hodnoty dobré a když je dost tréninkových dat.
- Problém: Jak stanovíme dobré iniciální hodnoty?

Algoritmus EM pro GMM – část 1

- 1 Stanov počáteční parametry $\hat{B}^0 = \{p(m), \mu_m, \Sigma_m\}$ pro všechna m .
- 2 Pro $i = 1, 2, 3, \dots$:
 - (1) Krok expectation: Stanov pro každou třídu Ω_m a každý příznakový vektor \mathbf{x}_j a posteriori pravděpodobnost:

$$\gamma_{j,m}^i = P(\Omega_m | \mathbf{x}_j, \hat{B}^{i-1}) = \frac{\mathcal{N}(\mathbf{x}_j | \mu_m^{i-1}, \Sigma_m^{i-1}) \cdot p(m)}{\sum_k \mathcal{N}(\mathbf{x}_j | \mu_k^{i-1}, \Sigma_k^{i-1}) \cdot p(k)}$$

Algoritmus EM pro GMM – část 2

(2) Maximization krok: Vypočítej nové parametry \hat{B}^i :

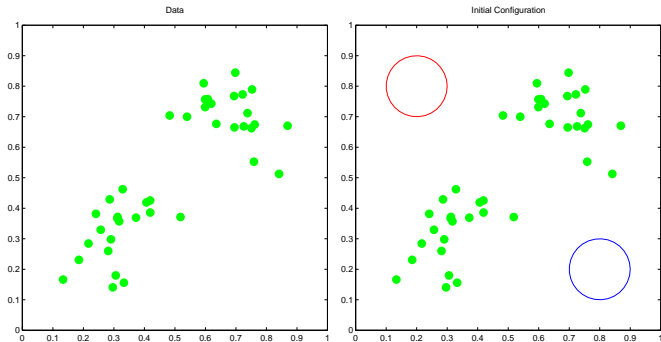
$$\rho(m) = \frac{1}{J} \sum_{j=1}^J \gamma_{j,m}^i$$

$$\mu_m = \frac{1}{\sum_j \gamma_{j,m}^i} \sum_{j=1}^J \gamma_{j,m}^i \cdot \mathbf{x}_j$$

$$\Sigma_m = \frac{1}{\sum_j \gamma_{j,m}^i} \sum_{j=1}^J \gamma_{j,m}^i \cdot (\mathbf{x}_j - \mu_m) \cdot (\mathbf{x}_j - \mu_m)^\top$$

(3) Kontroluj vhodnou terminační podmínku.

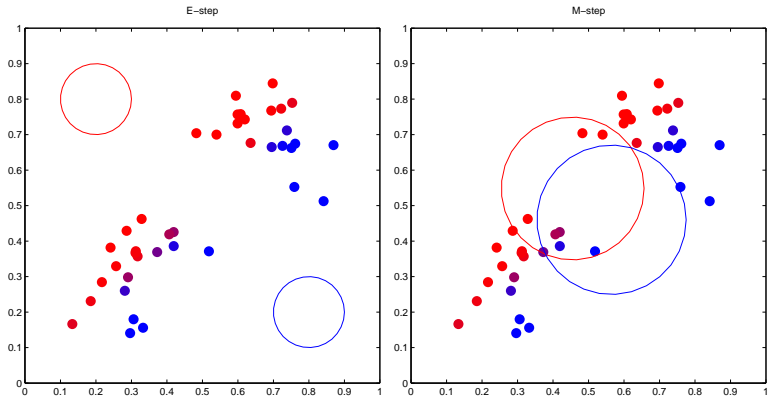
Algorithmus EM



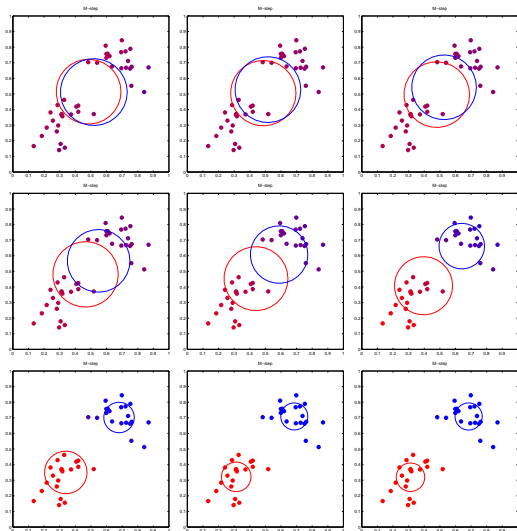
$$\mu_1 = \begin{pmatrix} 0,2 \\ 0,8 \end{pmatrix}, \quad \Sigma_1 = \begin{bmatrix} 0,01 & 0 \\ 0 & 0,01 \end{bmatrix}$$

$$\mu_2 = \begin{pmatrix} 0,8 \\ 0,2 \end{pmatrix}, \quad \Sigma_2 = \begin{bmatrix} 0,01 & 0 \\ 0 & 0,01 \end{bmatrix}$$

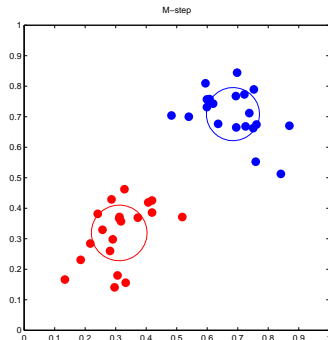
Algorithmus EM



Algorithmus EM



Algorithmus EM



$$\mu_1 = \begin{pmatrix} 0,31 \\ 0,32 \end{pmatrix}, \Sigma_1 = \begin{bmatrix} 0,008 & 0 \\ 0 & 0,008 \end{bmatrix}$$

$$\mu_2 = \begin{pmatrix} 0,68 \\ 0,71 \end{pmatrix}, \Sigma_2 = \begin{bmatrix} 0,008 & 0 \\ 0 & 0,008 \end{bmatrix}$$

Algoritmus EM*

- zjednodušená varianta algoritmu EM
- stanoví v iteračním kroku nejpravděpodobnější konstelaci \mathbf{u}^* neznámých dat s \mathbf{x} a θ
- domnívá se, že \mathbf{U} má opravdovou hodnotu \mathbf{u}^*
- odhaduje nové ML parametry θ^* vzhledem k $P(\mathbf{x}, \mathbf{u}^* | \cdot)$
- místo $Q(\hat{B}^j | \hat{B}^{j-1}) = \int_{\mathcal{Y}} P(Y|X, \hat{B}^{j-1}) \log P(X, Y | \hat{B}^j) dY$ bude maximalizováno $\mathcal{L}_{EM^*} = \max_{\mathcal{Y}} \log P(X, Y | \hat{B})$
- tj. tvoření podmíněné očekávané hodnoty je nahrazeno tvořením maxima
- výsledný algoritmus EM* už nemaximalizuje log-likelihood tréninkového vzorku jako algoritmus EM, ale jen účelovou veličinu \mathcal{L}_{EM^*}

Algoritmus EM*

- 1 Stanov vhodné počáteční parametry \hat{B}^0 .
- 2 Pro $i = 1, 2, 3, \dots$:
 - (1) Stanov nejpravděpodobnější hodnoty skrytých náhodných proměnných:

$$Y^i = \operatorname{argmax}_Y P(X, Y \mid \hat{B}^{i-1})$$

- (2) Vypočítej odhadní hodnoty pro kompletní data (X, Y^i) :

$$\hat{B}^i = \operatorname{argmax}_{\hat{B}} P(X, Y^i \mid \hat{B})$$

- (3) Kontroluj pomocí vhodné terminační podmínky.

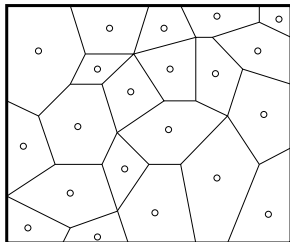
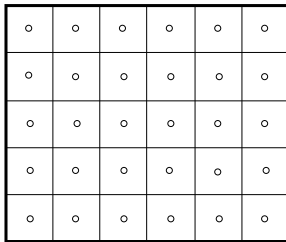
Algoritmus EM* — názorně

- maximalizuj $P(\mathbf{x}, \mathbf{u} \mid \cdot)$ místo $P(\mathbf{x} \mid \cdot)$
- data \mathbf{u} nejsou pozorovatelná
- přechod na podmíněnou očekávanou hodnotu vzhledem k a posteriori hustotě $P(\mathbf{u} \mid \mathbf{x}, \theta)$
- evaluace těchto s užíváním „starých“ parametrů distribuce θ předchozí iterace

Vektorová kvantizace

- Vektorová kvantizace je zobrazení libovolných příznakových vektorů \mathbf{x} na stanovené prototypové vektory \mathbf{z}_k .
- Množina prototypových vektorů $\mathcal{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_K\}$ tvoří **kódovou knihu** kvantizéru.
- Vektorový kvantizér stanoví rozklad vektorového prostoru, každá buňka se skládá ze všech příznakových vektorů, které jsou zobrazeny na stejný prototypový vektor.

dvě různé kvantizace dvojdimenzionálního vektorového prostoru:



Vektorový kvantizér

- Formálně: Vektorový kvantizér je operátor

$$q : \begin{cases} \mathbb{R}^D & \longrightarrow \mathcal{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_K\} \\ \mathbf{x} & \longmapsto q(\mathbf{x}) \end{cases}$$

- Zobrazení $q(\cdot)$ disjunktním rozkladem příznakového prostoru do buněk:

$$\mathbb{R}^D = \mathcal{Y}_1 \uplus \dots \uplus \mathcal{Y}_K, \quad \mathcal{Y}_\kappa = \{\mathbf{x} \mid q(\mathbf{x}) = \mathbf{z}_\kappa\}$$

- Optimální vektorový kvantizér $q(\cdot)$ minimalizuje očekávanou kvantizační chybu

$$\varepsilon = E[d(\mathbf{X}, q(\mathbf{X}))] = \sum_{\kappa=1}^K \int_{\mathbf{x} \in \mathcal{Y}_\kappa} d(\mathbf{x}, \mathbf{z}_\kappa) P(\mathbf{x}) d\mathbf{x} .$$

- d je přitom metrikou vzdálenosti mezi dvěma vektory.
- Problém: žádné uzavřené řešení pro optimální kvantizér

Optimální vektorový kvantizér

- Nutné podmínky pro kódovou knihu optimálního vektorového kvantizéru:
 - Kvantizér vybere nejbliž položený vektor z kódové knihy
 - $q(\cdot)$ rozdělí příznakový prostor do buněk $\hat{\mathcal{Y}}_1(\mathcal{Z}), \dots, \hat{\mathcal{Y}}_K(\mathcal{Z})$, kde $\hat{\mathcal{Y}}_\kappa(\mathcal{Z}) = \{\mathbf{x} \in \mathbb{R}^D \mid d(\mathbf{x}, \mathbf{z}_\kappa) = \min_\lambda d(\mathbf{x}, \mathbf{z}_\lambda)\}$.
 - Třídivý centroid $\mathbf{z}(\mathcal{Y}_\kappa)$ bude prototypovým vektorem \mathbf{z}_κ .
 - Třídivý centroid je vektor $\hat{\mathbf{y}}$ s minimální očekávanou odchylkou od prvků buňky

$$E[d(\mathbf{X}, \hat{\mathbf{y}}) \mid \mathbf{X} \in \mathcal{Y}_\kappa] = \min_{\mathbf{y}} E[d(\mathbf{X}, \mathbf{y}) \mid \mathbf{X} \in \mathcal{Y}_\kappa].$$

- Pro kvadratické míry vzdálenosti platí $d(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \mathbf{y})$:
Centroid množiny \mathcal{M} je podmíněná očekávaná hodnota

$$E[\mathbf{X} \mid \mathbf{X} \in \mathcal{M}].$$

Algoritmus k-means

- Vyber počet K tříd kódové knihy.
- Vyber iniciální prototypové vektory $\mathcal{Z}^{(0)} = (\mathbf{z}_{\kappa}^{(0)} \mid \kappa = 1, \dots, K)$.
- Pro $i = 1, 2, 3, \dots$:

- 1 Klasifikuj všechny tréninkové vektory $\mathbf{x} \in \omega$ a určuj z toho nový rozklad

$$\mathcal{Y}_{\kappa}^{(i)} = \hat{\mathcal{Y}}_{\kappa}(\mathcal{Z}^{(i-1)}), \quad \kappa = 1, \dots, K.$$

- 2 Vypočítej novou kódovou knihu $\mathcal{Z}^{(i)}$ s třídivými centroidy

$$\mathbf{z}_{\kappa}^{(i)} = \frac{1}{N_{\kappa}^{(i)}} \sum_{\mathbf{x} \in \mathcal{Y}_{\kappa}^{(i)}} \mathbf{x};$$

$N_{\kappa}^{(i)}$ je počet vektorů v buňce $\mathcal{Y}_{\kappa}^{(i)}$.

- 3 Když kritérium terminace je splněno \rightarrow konec
- Hledaná je kódová kniha $\mathcal{Z}^{(i)}$ a vektorový kvantizér rozhoduje podle pravidla minimální vzdálenosti.

Vlastnosti algoritmu k-means

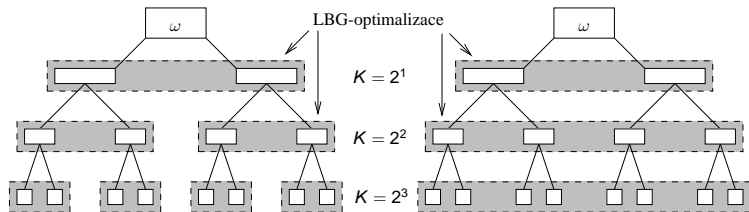
- víc tříd \rightarrow menší minimální kvantizační chyba
- každý K -kvantizér je také $(K + 1)$ -kvantizér
 \rightarrow žádná vhodná metoda výběru příznivé velikosti kódové knihy
- řízení ISODATA: rozšíření k-means; optimalizuje i počet tříd
- kvantizační chyba klesá monotónně a je pozitivní \rightarrow konvergence
- vhodné kritérium terminace: redukce chyby příliš malá
- kódová kniha je lokálně optimální
- iniciální kódová kniha určuje lokální minimum \Rightarrow metody na inicializaci kódové knihy
- pokryv algoritmu k-means náhodným procesem s klesající entropií (např. simulated annealing) \rightarrow globální optimálnost garantovaná, ale vyžaduje hodně času na výpočet

Algoritmus LBG (Linde, Buzo, Gray)

- hierarchická metoda pro snížení závislosti od inicializace
 1. Začni s triviálním rozkladem $\{\omega\}$ (celá tréninková množina je jedna buňka, $K = 1$).
 2. Rozlož K buňky heuristickou metodou do $2K$ buněk: najdi v každé buňce dva vzdálené body a použij je jako nové prototypové vektory.
 3. Urči algoritmem k-means iterativně optimální prototypový vektor pro novou kódovou knihu.
 4. Vrať se k bodu 2. dokud nebude dosaženo požadovaného počtu tříd.
- kombinace postupného rozkladu a postupného splynutí tříd kódové knihy, např. algoritmus ISODATA

Algoritmus LBG

- Optimalizaci k-means lze aplikovat v každém kroku buď na všechny nebo jen na část tříd kódové knihy.



- Pozor: V literatuře jsou názvy algoritmů LBG, k-means a Lloydovu algoritmu často poptené!
- Jsou vektorové kvantizéry, které užívají LBG.