Interakce člověk–počítač v přirozeném jazyce (ICP)

LS 2013 — Markovové modely, modelace fonů

Tino Haderlein, Elmar Nöth

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

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

Rozpoznávání jednotlivých slov

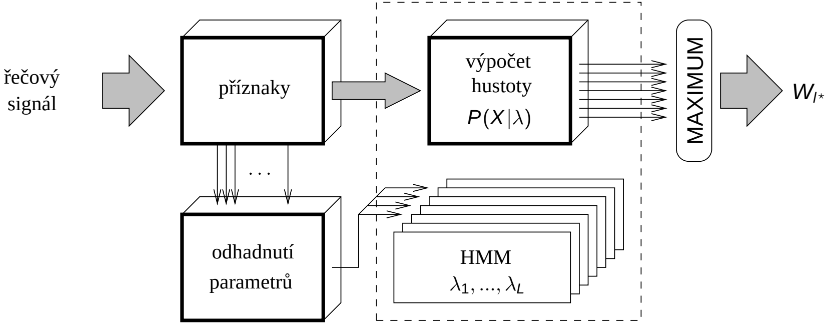
- Ja dána množina slov $\mathcal{W} = \{W_1, \ldots, W_L\}$.
- Je pozorována posloupnost příznakových vektorů $\boldsymbol{X} = \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$ (promluva).
- Které slovo bylo řečeno?

$\longrightarrow$ rozhodnutí podle Bayesovy věty

$$l^* = \operatorname*{argmax}_l P(W_l \mid \boldsymbol{X}), \quad \text{kde} \quad P(W_l \mid \boldsymbol{X}) = \frac{P(\boldsymbol{X} \mid W_l) \cdot P(W_l)}{P(\boldsymbol{X})}$$

- A priori pravděpodobnosti slov $P(W_l)$ se odhadují vypočítáním testovacího vzorku.
- $P(\boldsymbol{X} \mid W_l)$ nelze aproximovat hustotou normalního rozdělení, protože délka T promluvy je variabilní.
- Ale $P(\boldsymbol{X} \mid W_l)$ lze odhadnout pomocí skrytých Markovových modelů (Hidden Markov Models, HMM).

Rozpoznávání jednotlivých slov



Hidden Markov Models

Úloha: Vypočítej pravděpodobnost $P(x_1, x_2, ..., x_T)$ pro libovolné hodnoty délky T.

- Problém 1: Náhodný proces, který tvoří x_t, ve skutečnosti není stacionární, tj. $P(x_t)$ je např. závislé na mluvené hlásce; proto není možné snadně počítat $P(x_1, x_2, .., x_T) = \prod_{t=1}^{T} P(x_t)$.
- Problém 2: x_t mohou být příznakové vektory, proto je např. počítání $P(x_{t-1}, x_t)$ náročné; když má x_t 24 dimenzí, jsou pro 10 příznakových vektorů nutné 240-dimenzionální hustoty.

Hidden Markov Models

Trik 1: Uvedení diskrétních, skrytých stavů $q_t \in \{s_1, ..., s_N\}$

- x_t jsou závislé jen na aktuálním stavu q_t:
  $P(x_1, x_2, ..., x_T) =$
  $\sum_{q_1,...,q_T} P(q_1) \cdot P(x_1|q_1) \cdot \prod_{t=2}^{T} P(x_t|q_t) \cdot P(q_t|q_1, ...q_{t-1})$

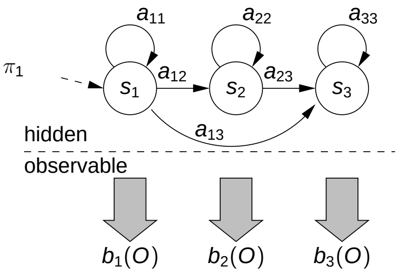
Trik 2: Předpokládej, že náhodný proces, který tvoří q_t, je stacionární Markovův proces prvního řádu, tj. platí
$P(q_t \mid q_1, ..., q_{t-1}) = P(q_t \mid q_{t-1})$.

- Pak lze vypočítat $P(x_1, x_2, ..., x_T)$ takto:
  $P(x_1, x_2, .., x_T) = \sum_{q_1,..,q_T} P(q_1) \cdot P(x_1|q_1) \cdot \prod_{t=2}^{T} P(x_t|q_t) \cdot P(q_t|q_{t-1})$

Pro vypočítání $P(x_1, x_2, ..., x_T)$ jsou tedy nutné jen hustoty $P(x_t|q_t), q_t \in \{s_1, ..., s_N\}$ a pravděpodobnosti $P(q_t|q_{t-1})$.

- Pokud je Markovův proces stacionární, $P(q_t|q_{t-1})$ jsou nezávislé na t; lze je uložit do matice NxN.
- $P(q_1)$ jsou N různé hodnoty.
- Lze vybrat $P(x_t|q_t)$ podle aplikace, např. N různých hustot normálního rozdělení.

Hidden Markov Models: definice



HMM $\boldsymbol{\lambda} = (\boldsymbol{\pi}, \mathbf{A}, \boldsymbol{B})$, kde

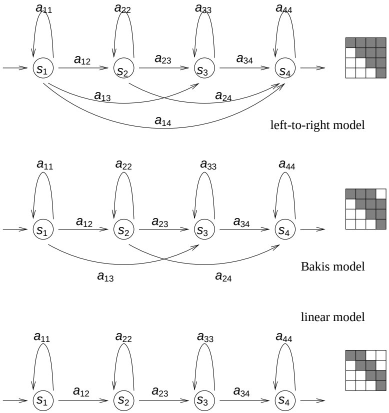
- $\boldsymbol{\pi} = (\pi_i)$: $P(q_1)$ počáteční pravděpodobnosti
- $\boldsymbol{A} = [a_{ij}]$: $P(q_t = s_j | q_{t-1} = s_i)$ přechodové pravděpodobnosti
- $\boldsymbol{B} = (b_j)$: $P(x_t | q_t = s_j)$ výstupní pravděpodobnosti podle typu modelu:
  - diskrétní HMM: konečná výstupní abeceda, např. třídy kódové knihy vektorového kvantizéru
  - kontinuální HMM: vážené sumy rozdělení
    $b_j(\boldsymbol{x}) = \sum_{k=1}^{K_j} \omega_{jk} \cdot \mathcal{N}(\boldsymbol{x} | \boldsymbol{\mu}_{jk}, \boldsymbol{\Gamma}_{jk})$
  - semi-kontinuální HMM:
    $b_j(\boldsymbol{x}) = \sum_{k=1}^{K} \omega_{jk} \cdot \mathcal{N}(\boldsymbol{x} | \boldsymbol{\mu}_{k}, \boldsymbol{\Gamma}_{k})$

HMM: Základní otázky

- Jak bychom měli stanovit topologie HMM, tj. které přechody mezi stavy bychom měli dovolit a které ne, tj. $P(s_i|s_j) = 0$?
- Jak vypočítáme produkční pravděpodobnost $P(x_1, .., x_T|\lambda)$ účinně?
- Dekódování: Jak získáme nejpravděpodobnější posloupnost stavů, když je dáno pozorování $x_1, ..x_T$?
- Jak odhadujeme parametry HMM z tréninkového vzorku?

Topologie HMM

- topologie a počet stavů se většinou stanoví manuálně
- vhodné modelové topologie pro rozpoznávání řeči;
  prvky přechodové matice, které nejsou nula, jsou vybarveny šedivě



- HMM, ve kterém jsou spojené všechny stavy, se jmenuje ergodický HMM.

Určení produkční pravděpodobnosti

Hledáme $P(\boldsymbol{O} \mid \boldsymbol{\lambda})$, tj. pravděpodobnost, že $\boldsymbol{O} = O_1, ..., O_T$ je tvořeno pomocí $\boldsymbol{\lambda}$.

- jako nahoře, jen v nové notaci: počítání produkční pravděpodobnosti sumací všech možných posloupností stavů $P(\boldsymbol{O} \mid \boldsymbol{\lambda}) = \sum_{\boldsymbol{q} \in \mathcal{Q}^T} P(\boldsymbol{O}, \boldsymbol{q} \mid \boldsymbol{\lambda}) = \sum_{\boldsymbol{q} \in \mathcal{Q}^T} \pi_{q_1} b_{q_1}(O_1) \cdot \prod_{t=2}^{T} a_{q_{t-1} q_t} b_{q_t}(O_t)$
- kolem $2T \cdot N^T$ násobení: exponenciální komplexita s T
- zjednodušení počítání zavedením dopředných a zpětných pravděpodobností α, β
- dopředná pravděpodobnost: $\alpha_t(j) = P(O_1 \ldots O_t, q_t = j \mid \boldsymbol{\lambda})$
- zpětná pravděpodobnost: $\beta_t(i) = P(O_{t+1} \ldots O_T \mid q_t = i, \boldsymbol{\lambda})$
- v každé chvíli t platí: $\alpha_t(j) \cdot \beta_t(j) = P(\boldsymbol{O}, q_t = j \mid \boldsymbol{\lambda}), \quad P(\boldsymbol{O} \mid \boldsymbol{\lambda}) = \sum_{j=1}^{N} \alpha_t(j) \beta_t(j)$

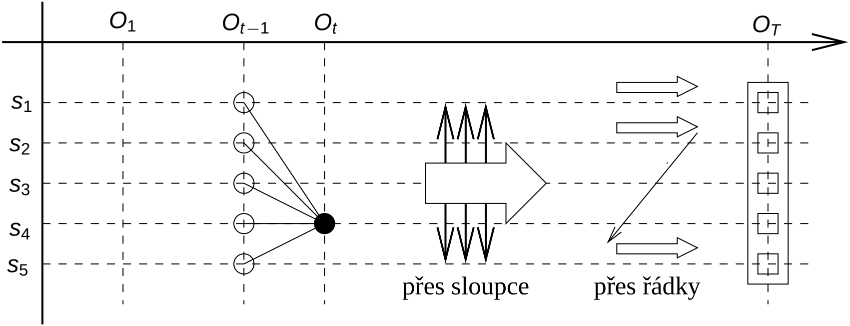
Určení produkční pravděpodobnosti

dva ekvivalentní algoritmy, buď s dopřednými anebo zpětnými pravděpodobnostmi

- **Inicializace**:
  Pro všechna $j = 1, \ldots, N$ buď
  $$\alpha_1(j) = \pi_j b_j(O_1)$$
  Pro všechna $i = 1, \ldots, N$ buď
  $$\beta_T(i) = 1$$
- **Rekurze**:
  Pro $t > 1$ a $j = 1, \ldots, N$ buď
  $$\alpha_t(j) = \left(\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}\right) b_j(O_t)$$
  Pro $t < T$ a $i = 1, \ldots, N$ buď
  $$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$$
- **Terminace**:
  Počítej
  $$\mathrm{P}(\boldsymbol{O} \mid \boldsymbol{\lambda}) = \sum_{j=1}^{N} \alpha_T(j)$$
  Počítej
  $$\mathrm{P}(\boldsymbol{O} \mid \boldsymbol{\lambda}) = \sum_{j=1}^{N} \pi_j b_j(O_1) \beta_1(j)$$

Určení produkční pravděpodobnosti

- Oba algoritmy mají stejnou výpočetní složitost.
- Komplexita je kvadratická vzhledem k N a lineární v T: $2 \cdot N^2 \cdot T$ násobení



Viterbiův algoritmus

Hledáme posloupnost stavů $q_1, ..., q_T$, která maximalizuje $P(q_1, ..., q_T \mid \boldsymbol{O}, \boldsymbol{\lambda})$, tj. posloupnost stavů, kterou HMM při pozorování nejpravděpodobneji probíhal.

- **a posteriori pravděpodobnost** vztažená na výstup $P(\boldsymbol{q} \mid \boldsymbol{O}, \boldsymbol{\lambda}) = \frac{P(\boldsymbol{O}, \boldsymbol{q} | \boldsymbol{\lambda})}{P(\boldsymbol{O} | \boldsymbol{\lambda})}$
- $\boldsymbol{q}^*$ je **optimální posloupnost stavů**, když $P(\boldsymbol{O}, \boldsymbol{q}^* \mid \boldsymbol{\lambda}) = \max_{\boldsymbol{q} \in \mathcal{Q}^T} P(\boldsymbol{O}, \boldsymbol{q} \mid \boldsymbol{\lambda}) =: P^*(\boldsymbol{O} \mid \boldsymbol{\lambda})$.
- **Viterbiův algoritmus:** varianta počítání dopředné matice
- místo $\alpha_t(j)$ se počítají tyto pravděpodobnosti: $\vartheta_t(j) = \max\{P(O_1 \dots O_t, q_1 \dots q_t \mid \boldsymbol{\lambda}) \mid \boldsymbol{q} \in \mathcal{Q}^T\},$ kde $q_t = j$
- zpětné indexy pro extrakci posloupnosti stavů

Viterbiův algoritmus

- **Inicializace**:
  Buď $\vartheta_1(j) = \pi_j b_j(O_1)$ a $\psi_1(j) = 0$ pro všechny $j = 1, \ldots, N$.
- **Rekurze**: Pro všechny $j = 1, \ldots, N$ buď

$$\vartheta_t(j) = \max_i (\vartheta_{t-1}(i) a_{ij}) b_j(O_t); \quad \psi_t(j) = \operatorname*{argmax}_i \vartheta_{t-1}(i) a_{ij}$$

- **Terminace**: Buď

$$\mathrm{P}^*(\boldsymbol{O} \mid \boldsymbol{\lambda}) = \max_j \vartheta_T(j); \quad q_T^* = \operatorname*{argmax}_j \vartheta_T(j)$$

- **Zpětné sledování**: Pro $t = T - 1, \ldots, 1$ lze počítat optimální posloupnost pomocí

$$q_t^* = \psi_{t+1}(q_{t+1}^*).$$

Odhad ML modelových parametrů

Hledáme množinu $\hat{\lambda}$ parametrů HMM, když je dána tréninková promluva $\boldsymbol{O}$.

- odhad ML: Vyber $\hat{\lambda}$ tak, aby cílová funkce byla maximální: $\log_{\text{HMM}}(\boldsymbol{\lambda}) = \log P(\boldsymbol{O} \mid \boldsymbol{\lambda}) = \log \sum_{\boldsymbol{q} \in \mathcal{Q}^T} P(\boldsymbol{O}, \boldsymbol{q} \mid \boldsymbol{\lambda})$
- pozorovatelná náhodná proměnná: $X = \boldsymbol{O}$
- skrytá náhodná proměnná: $Y = \boldsymbol{q}$
- parametr, který odhadujeme: $B = \hat{\boldsymbol{\lambda}}$
- vztah mezi $\boldsymbol{O}$ a $\boldsymbol{q}$ je znám
  $\rightarrow$ aplikace algoritmu EM
- maximalizuj $\hat{\lambda}$ vzhledem ke **Kullback-Leiblerově statistice** $Q(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) = \sum_{\boldsymbol{q} \in \mathcal{Q}^T} P(\boldsymbol{q} \mid \boldsymbol{O}, \boldsymbol{\lambda}) \cdot \log P(\boldsymbol{O}, \boldsymbol{q} \mid \hat{\boldsymbol{\lambda}})$

Odhad ML modelových parametrů

Pro počítání a posteriori pravděpodobnosti $P(\boldsymbol{q} \mid \boldsymbol{O}, \boldsymbol{\lambda})$ se uvádí následující veličiny:

- **a posteriori přechodná pravděpodobnost** pro $s_i \rightarrow s_j$ ve chvíli t:

$$\begin{aligned} \xi_t(i,j) &= P(q_t = i, q_{t+1} = j \mid \boldsymbol{O}, \boldsymbol{\lambda}) \\ &= \frac{P(q_t = i, q_{t+1} = j, \boldsymbol{O} \mid \boldsymbol{\lambda})}{P(\boldsymbol{O} \mid \boldsymbol{\lambda})} \\ &= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \alpha_t(i) \beta_t(i)}, \quad 1 \leq t < T \end{aligned}$$

- **a posteriori stavová pravděpodobnost** pro s_i ve chvíli t:
  $\gamma_t(i) = P(q_t = i \mid \boldsymbol{O}, \boldsymbol{\lambda}) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} = \sum_{j=1}^{N} \xi_t(i,j)$
- sumace těch $\xi_t(i,j)$, popř. $\gamma_t(i)$ přes všechna t
  $\rightarrow$ očekávané hodnoty pro přechody $s_i \rightarrow s_j$, popř. pro pobyt v s_i

Baum-Welchovy rovnice

Rovnice odhadu, které vznikají z algoritmu EM pro HMM, se nazývají Baum-Welchovy rovnice; trénink se nazývá Baum-Welchův trénink nebo algoritmus prohledávání dopředu a dozadu.

$$\begin{aligned}
\hat{\pi}_i &= \gamma_1(i) = \frac{\alpha_1(i)\beta_1(i)}{\sum_{j=1}^{N} \alpha_1(j)\beta_1(j)} \\
\hat{a}_{ij} &= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i)\beta_t(i)} \\
\hat{b}_{jk} &= \frac{\sum_{t=1}^{T} \gamma_t(j) \chi_{[O_t=v_k]}}{\sum_{t=1}^{T} \gamma_t(j)} = \frac{\sum_{t=1}^{T} \alpha_t(j)\beta_t(j) \chi_{[O_t=v_k]}}{\sum_{t=1}^{T} \alpha_t(j)\beta_t(j)}
\end{aligned}$$

- $\chi_{[\cdot]} = 1$ pro případné výroky a jinak 0.
- Iterativní aplikace vede k lokálnímu optimu.
- Nárok jedné iterace je jen o málo vyšší než počítání dopředné a zpětné matice.

Viterbiův trénink

- Aplikace algoritmu EM* se nazývá Viterbiův trénink nebo segmentově členěný algoritmus k-means.
- Optimalizuje se podle Viterbiova hodnocení $P^*(\boldsymbol{O} \mid \boldsymbol{\lambda}) = P(\boldsymbol{O}, \boldsymbol{q}^* \mid \boldsymbol{\lambda}^{(n-1)})$.
- Je **mnohem účinnější** než Baum-Welchův trénink (BWT).
- Při malém tréninkovém vzorkům **není spolehlivější** než BWT.
- Viterbiův trénink odpovídá BWT s **modifikovanými a posteriori pravděpodobnostmi:** $\gamma_t^*(i) = \chi_{[q_t^*=s_i]}, \quad \xi_t^*(i,j) = \chi_{[q_t^*=s_i, q_{t+1}^*=s_j]}$

Viterbiův trénink

- Vyber počáteční model $\boldsymbol{\lambda}^{(0)}$.
- Pro $n = 1, 2, \ldots$:
  (1) Urči optimální posloupnost stavů $\boldsymbol{q}^*$, kde

$$\mathrm{P}(\boldsymbol{O}, \boldsymbol{q}^* \mid \boldsymbol{\lambda}^{(n-1)}) = \max_{\boldsymbol{q}} \mathrm{P}(\boldsymbol{O}, \boldsymbol{q} \mid \boldsymbol{\lambda}^{(n-1)})$$

  pomocí Viterbiova algoritmu.
  (2) Spočítej počáteční, přechodové a výstupní četnosti patřící ke $\boldsymbol{q}^*$

$$\bar{\pi}_i = \chi_{[q_1 = s_i]}\,, \quad \bar{a}_{ij} = \sum_{t=1}^{T-1} \chi_{[q_t^* = s_i, q_{t+1}^* = s_j]}\,, \quad \bar{b}_{jk} = \sum_{t=1}^{T} \chi_{[q_t^* = s_j, O_t = v_k]}\,.$$

  (3) Normalizuj $\hat{\pi}_i = \bar{\pi}_i / \sum_i \bar{\pi}_i$, $\quad \hat{a}_{ij} = \bar{a}_{ij} / \sum_j \bar{a}_{ij}$, $\quad \hat{b}_{jk} = \bar{b}_{jk} / \sum_k \bar{b}_{jk}$.
  (4) Stanov $\boldsymbol{\lambda}^{(n)} = (\hat{\pi}_i, \hat{a}_{ij}, \hat{b}_{jk})$.

Praktická aplikace skrytých Markovových modelů

Pro dlouhé promluvy při omezené numerické přesnosti může násobení mnohých pravděpodobností rychle vést k hodnotám $= 0$.

- 1. řešení: V algoritmu prohledávání dopředu užíváme stupnice/normalizace
  $\tilde{\alpha}_t(j) \;=\; \frac{1}{C_t} \cdot \alpha_t(j) \;=\; \frac{\alpha_t(j)}{\sum_i \alpha_t(i)}$
  s faktory stupnice C_t vztažené na čas t.
  $\alpha_t(j)$ můžeme vždycky rekonstruovat, protože platí
  $\alpha_t(j) = C_1 \cdot C_2 \cdots C_t \cdot \tilde{\alpha}_t(j)$.
- 2. řešení: logaritmizace:
  - Ve Viterbiovu algoritmu pracujeme s logaritmovanými pravděpodobnostmi $\rightarrow$ násobení se stane adicí.
  - Problém: V algoritmu prohledávání dopředu musíme sčítat logaritmované pravděpodobnosti $\rightarrow$ **Kingsbury-Raynerova rovnice** $\log(p_1 + p_2) \;=\; \log p_1 + \log(1 + e^{\log p_2 - \log p_1})$.
  - zrychlení s tabulkou všech $\log(1 + e^{\log p_2 - \log p_1})$

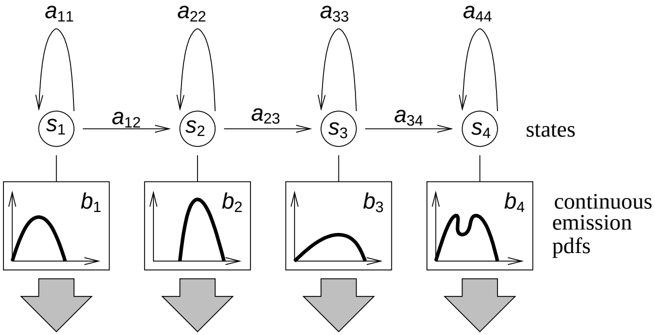
Praktická aplikace skrytých Markovových modelů

- Je nutné vhodné stanovení iniciálních odhadnutých hodnot $\boldsymbol{\lambda}^{(0)}$:
  - S algoritmem LBG shlukujeme (clustering) všechny příznakové vektory; odhadujeme iniciální diskrétní HMM, který je inicializací kontinuálního HMM.
  - Výstupní rozdělení kontinuálního HMM inicializujeme prostřednictvím shluku segmentace, kterou vytvořil diskrétní HMM.
- Když je pro HMM víc než jeden tréninkový příklad, rovnice na odhad se mění jako při algoritmu EM.
- Příklad: Rovnice na odhad přechodné pravděpodobnosti:

$$\hat{a}_{ij} = \frac{\sum_{l=1}^{L} \sum_{m=1}^{M_l} \left(\sum_{t=1}^{T_{l,m}-1} \xi_t^{(l,m)}(i,j) \right)}{\sum_{l=1}^{L} \sum_{m=1}^{M_l} \left(\sum_{t=1}^{T_{l,m}-1} \gamma_t^{(l,m)}(i) \right)}$$

Kontinuální HMM

- Diskrétní HMM vyžadují předchozí vektorovou kvantizaci: $\boldsymbol{X} = \boldsymbol{x}_1 \ldots \boldsymbol{x}_T \longrightarrow \boldsymbol{O} = O_1, \ldots, O_T \Rightarrow$ **ztráta informace**
- **Kontinuální výstupní rozdělovací hustoty** $b_j(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^D$ pro zpracování příznakových vektorů: chování charakterizováno váženou sumou hustot rozdělení $P(\boldsymbol{X}, \boldsymbol{q} \mid \boldsymbol{\lambda})$



- $b_j(\boldsymbol{x})$ z **parametrické rodiny** hustot, např. normální hustota $\mathcal{N}$

Kontinuální HMM: Rovnice na odhad

- Rovnice na odhad počátečních a přechodových pravděpodobností a počítání hodnot pravděpodobností jsou ekvivalentní s diskrétními HMM.
- Při normálních rozdělovacích hustotách platí

$$b_j(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$$

$$\begin{aligned}
\hat{\boldsymbol{\mu}}_j &= \frac{1}{\sum_t \gamma_t(j)} \sum_{t=1}^{T} \gamma_t(j) \boldsymbol{x}_t \\
\hat{\boldsymbol{\Sigma}}_j &= \frac{1}{\sum_t \gamma_t(j)} \sum_{t=1}^{T} \gamma_t(j)(\boldsymbol{x}_t - \hat{\boldsymbol{\mu}}_j)(\boldsymbol{x}_t - \hat{\boldsymbol{\mu}}_j)^{\mathrm{T}} \\
&= \frac{1}{\sum_t \gamma_t(j)} \sum_{t=1}^{T} \gamma_t(j) \boldsymbol{x}_t \boldsymbol{x}_t^{\mathrm{T}} - \hat{\boldsymbol{\mu}}_j \hat{\boldsymbol{\mu}}_j^{\mathrm{T}}.
\end{aligned}$$

Gaußovy vážené sumy hustot rozdělení

- Výstupní hustota je Gaußova vážená suma hustot rozdělení (svázaná/spojená hustota rozdělení):

$$b_j(\boldsymbol{x}) = \sum_{k=1}^{K} c_{jk} g_{jk}(\boldsymbol{x}) = \sum_{k=1}^{K} c_{jk} \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_{jk}, \boldsymbol{\Sigma}_{jk}) \,, \quad \sum_{k=1}^{K} c_{jk} = 1$$

- může při vysokém počtu hustot aproximovat každou hustotu
- rozšiřovaný v rozpoznávání řeči
- komponenta $k_t \in \mathcal{K}$ sumy je skrytá náhodná proměnná
- produkční pravděpodobnost

$$P(\boldsymbol{X} \mid \boldsymbol{\lambda}) = \sum_{\boldsymbol{q} \in \mathcal{Q}^T} \sum_{\boldsymbol{k} \in \mathcal{K}^T} P(\boldsymbol{X}, \boldsymbol{q}, \boldsymbol{k} \mid \boldsymbol{\lambda})$$

Baum-Welchovy rovnice odhadu

pro Gaußovy vážené sumy hustot rozdělení

a posteriori pravděpodobnost selekce komponenty k v s_j ve chvíli t:

$$\begin{aligned}\zeta_t(j,k) &= P(q_t = j, k_t = k \mid \boldsymbol{X}, \boldsymbol{\lambda}) \\ &= \begin{cases} \frac{1}{P(\boldsymbol{X}|\boldsymbol{\lambda})} \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} c_{jk} g_{jk}(\boldsymbol{x}_t) \beta_t(j), & \text{když } t > 1 \\ \frac{1}{P(\boldsymbol{X}|\boldsymbol{\lambda})} \sum_{i=1}^{N} \pi_j c_{jk} g_{jk}(\boldsymbol{x}_1) \beta_1(j) \quad\quad , & \text{když } t = 1 \end{cases}\end{aligned}$$

rovnice odhadu:

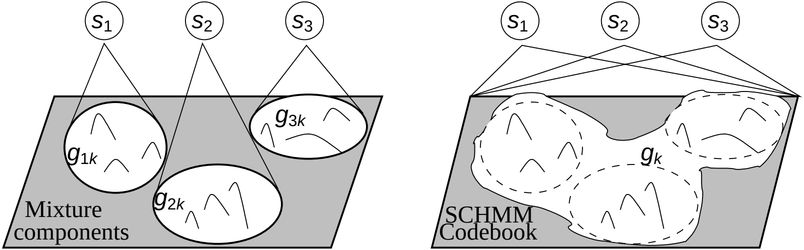
$$\begin{aligned}\hat{c}_{jk} &= \frac{1}{\sum_t \gamma_t(j)} \sum_{t=1}^{T} \zeta_t(j,k) \\ \hat{\boldsymbol{\mu}}_{jk} &= \frac{1}{\sum_t \zeta_t(j,k)} \sum_{t=1}^{T} \zeta_t(j,k) \boldsymbol{x}_t \\ \hat{\boldsymbol{\Sigma}}_{jk} &= \frac{1}{\sum_t \zeta_t(j,k)} \sum_{t=1}^{T} \zeta_t(j,k) \boldsymbol{x}_t \boldsymbol{x}_t^{\mathrm{T}} - \hat{\boldsymbol{\mu}}_{jk} \hat{\boldsymbol{\mu}}_{jk}^{\mathrm{T}}\end{aligned}$$

Semikontinuální HMM (SCHMM)

- Markovovy modely se **semikontinuálními** výstupními hustotami:

$$b_j(\boldsymbol{x}) = \sum_{k=1}^{K} c_{jk} g_k(\boldsymbol{x}) = \sum_{k=1}^{K} c_{jk} \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \quad \sum_{k=1}^{K} c_{jk} = 1$$

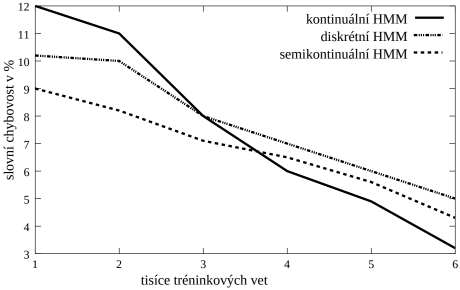
- rozdíl oproti kontinuálnímu HMM s váženými sumami hustot: tam komponenty pro **jeden** stav, tady pro **všechny** stavy
- SCHMM má schopnost aproximace jako vážené sumy hustot, vyžaduje ale méně parametrů.



Semikontinuální HMM

- Váhy c_{ij} sumy mohou být také považované za výstupní pravděpodobnosti diskrétního HMM.
- SCHMM hodnotí **všechny** třídy kódové knihy, hodnoty $g_k(\boldsymbol{x})$ hustoty váží diskrétní výstupní pravděpodobnost (**měkká** nebo **neostrá** vektorová kvantizace).
- Vektorová kvantizace je součástí SCHMM.
- SCHMM může tedy být zařazen mezi diskrétní a kontinuální HMM.

Vlastnosti semikontinuálních HMM



- Vlastnosti SCHMM:
  - kompaktní parametrový prostor
  - žádné zkřivení kvůli kvantizaci
  - vektorová kvantizace je zahrnuta do procesu optimalizace modelu
- SCHMM je lepší než kontinuální HMM, když je málo dat k dispozici; když je víc dat, je horší.

Semikont. HMM: Baum-Welchovy rovnice odhadu

sumace statistik hustot přes všechny stavy

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{\sum_t \sum_j \zeta_t(j,k)} \sum_{t=1}^{T} \sum_{j=1}^{N} \zeta_t(j,k) \boldsymbol{x}_t$$

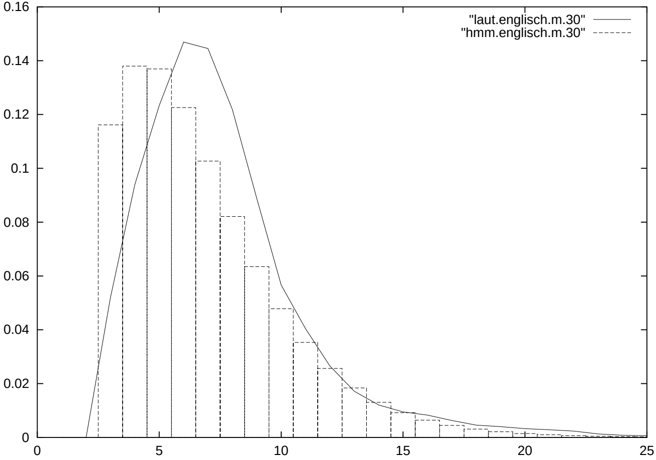
$$\hat{\boldsymbol{\Sigma}}_k = \frac{1}{\sum_t \sum_j \zeta_t(j,k)} \sum_{t=1}^{T} \sum_{j=1}^{N} \zeta_t(j,k) \boldsymbol{x}_t \boldsymbol{x}_t^{\mathrm{T}} - \hat{\boldsymbol{\mu}}_k \hat{\boldsymbol{\mu}}_k^{\mathrm{T}}$$

Nevýhody užívání HMM

- modelování trvání
  - Doba pobytu v **jednotlivém** stavu je rozdělená exponenciálně: $d_i(\tau) = P(q_{t-\tau+1} = \ldots = q_{t-1} = s_i, q_t \neq s_i \mid q_{t-\tau} = s_i,\ \boldsymbol{\lambda}) = a_{ii}^{\tau-1} \cdot (1 - a_{ii})$
  - Ve skutečnosti je ale nejpravděpodobnější délka hlásky > 0.
  - Řešení: explicitní modelování délky anebo vhodné modelové topologie s několikrát za sebou zapojenými stavy
  - Víc stavů může opravdové rozdělení délky hlásky aproximovat lépe.

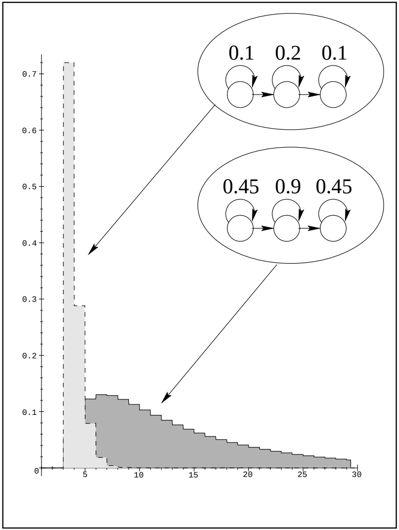
Nevýhody užívání HMM

rozdělení trvání anglické hlásky („laut“) a jeho model podle HMM



Nevýhody užívání HMM

různá modelování trvání hlásky pomocí HMM (x: doba; y: pravděpodobnost)



Nevýhody užívání HMM

- Markovův proces prvního řádu
  - řešení: generalizované HMM vyššího řádu
  - problém: vysoká výpočetní složitost
- předpoklad podmíněné nezávislosti pozorování
  - příklad: odhad diskrétního HMM s 2 stavy na textech `mmmmmmmmmm`, `wwwwwwwwww`.
  - srovnatelné s mužskými a ženskými mluvčími s váženými sumami rozdělení jako výstupní hustota
  - HMM bude mít vysoké produkční pravděpodobnosti pro texty `wwwwmmmm` a `mmmmwwww`.
  - Proč?
  - řešení: paralelně zapojené HMM, segmentové modely

HMM – střední hodnoty výstupní rozdělení