

# Úvod do počítačového zpracování řeči

Luděk Bártek

Fakulta informatiky  
Masarykova univerzita

podzim 2015

# Obsah

- 1 Rozpoznávání řeči
  - Rozpoznávání izolovaných slov
  - Rozpoznávání plynulé promluvy

# Cíle rozpoznávání řeči

- Cíle rozpoznávání řeči:
  - Interpretace příkazů uživatele – hlasové ovládání různých zařízení.
    - telefon
    - navigace
    - ...
  - Převod mluveného slova na text – přepis mluveného slova
    - záznamy soudních přelíčení
    - ...
- Druhy rozpoznávání řeči:
  - rozpoznávání izolovaných slov
  - rozpoznávání plynulé promluvy.

# Rozpoznávání izolovaných slov

## Úvod

- Cíl – rozpoznání částí promluvy ohraničených z obou stran pauzou.
- Uživatel může zadávat pouze jednotlivé povely nebo musí po vyřčení slova udělat pauzu.
- Odpadá problém se stanovením rozhraní dvou slov/povelů.
  - Povel může být víceslovný, ale pro tyto účely představuje jedno slovo.
- Obvykle jde o systémy závislé na uživateli
  - nutnost tréninku.
- Mívají omezenou kapacitu slovníku
  - slovník – seznam rozpoznávaných slov.
- Používají obvykle vektor příznaků.
  - Vektor hodnot získaných analýzou signálu (spektrum, kepstrum, energie, intenzity, ...).
  - Získán některou z metod krátkodobé analýzy.

# Vektory příznaků a jejich porovnávání

- Vektor příznaků
- Vektorový prostor nad tělesem  $F$  je množina  $V$  společně s dvěma operacemi sčítání vektorů a násobení skalárem, které splňují následující axiomy:
  - $(V, +)$  je komutativní grupa
  - Násobení skalárem ( $F \times V \rightarrow V$ ) je asociativní a  $a(b\mathbf{v}) = ab(\mathbf{v})$
  - $1\mathbf{v} = \mathbf{v}$ , kde  $1$  je jednotkový prvek tělesa
  - a dále platí distributivní zákon:
    - $a(\mathbf{v} + \mathbf{w}) = a\mathbf{v} + a\mathbf{w}$
    - $(a+b)\mathbf{v} = a\mathbf{v} + b\mathbf{v}$
- Metrický prostor: Množina  $M$  se zobrazením  $d$  (metrikou), pro které platí:
  - $d(x, y) \geq 0$
  - $d(x, y) = 0 \Leftrightarrow x = y$
  - $d(x, y) = d(y, x)$
  - $d(x, z) \leq d(x, y) + d(y, z)$
- Příklad metriky je např. Euklidovská vzdálenost.

# Klasifikátory

- Klasifikátory využívající porovnání slov metodou DTW (Dynamic Time Warping)
  - umožňují porovnání podobnosti dvou dynamických jevů, které probíhají různými rychlostmi.
- Klasifikátory založené na statistických metodách
  - modelování pomocí skrytých Markovových modelů.
- Hierarchické klasifikátory
  - Pracují hierarchicky:
    - 1 Akustická analýza signálu.
    - 2 Rozdělení signálu promluvy na segmenty.
    - 3 Fonetické dekódování jednotlivých segmentů.
    - 4 Rozpoznání slova (povelu) probíhá ve druhé vyšší úrovni na základě posloupnosti klasifikovaných segmentů.
  - Podobný princip se využívá pro rozpoznávání plynulé řeči.

# Metoda DTW (Borcení časové osy)

- Používá se pro porovnání dvou úseků promluv (slov).
  - Úseky jsou vyjádřeny posloupností vektorů příznaků
    - úsek promluvy rozdělen do mikrosegmentů
    - klasifikovány souborem krátkodobých charakteristik
- Postup:
  - 1 Pro rozpoznávané posloupnosti vytvoříme soubor referenčních posloupností akustických vektorů.
  - 2 Vytvoříme posloupnost akustických vektorů pro rozpoznávané slovo.
  - 3 Metodou DTW porovnáme rozpoznávanou posloupnost s referenčními a vybereme tu, s největší shodou.

# Metoda DTW

## Pokračování

- Algoritmus hledá parametrizaci  $f, g$  takovou, že  $i=f(k), j=g(k)$ ,  $k=1, \dots, K$ , minimalizuje výraz:

$$D(A, B) = \sum_{k=1}^K d(a(f(k)), b(g(k)))$$

- $d$  je vzdálenost mezi akustickými vektory (např. Euklidovská metrika)
- Euklidovská metrika

$$d(a, b) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}$$



# Metoda DTW (2.)

- Omezující podmínky:
  - $f, g$  – neklesající funkce
  - omezení na lokální souvislost a strmost:
    - $0 \leq f(k) - f(k-1) \leq I^*$
    - $0 \leq g(k) - g(k-1) \leq J^*$
    - většinou platí  $I^*, J^* = 1, 2, 3$
    - Z praktických testů vyplynulo, že při příliš strmém přírůstku může dojít např. k nevhodné korespondenci mezi příliš krátkým segmentem vzorku A a příliš dlouhým segmentem vzorku B.
  - Omezení na hraniční body:
    - $f(1) = 1, f(K) = I$
    - $g(1) = 1, g(K) = J$

# Metoda DTW (3.)

- Omezující podmínky
  - Globální vymezení oblasti pohybu funkce DTW:
    - Omezení minimální a maximální přípustné směrnice přímky omezující přípustnou oblast, při splnění podmínky na hraniční body
$$1 + \alpha[i(k) - 1] \leq j(k) \leq 1 + \beta[i(k) - 1]$$
    - $\alpha$  – minimální směrnice přímky omezující přípustnou oblast
    - $\beta$  – maximální směrnice přímky omezující přípustnou oblast
  - ...

# DTW – praktická realizace klasifikátoru slov

## Blokové schéma



Figure: Blokové schéma algoritmu DTW

# DTW – praktická realizace klasifikátoru slov

## Trénování

- Obecný postup:
  - 1 Řečník resp. skupina řečníků vysloví postupně každé trénované slovo požadovaného slovníku. Buď jednou nebo opakovaně
  - 2 Vstupní slova jsou zdigitalizována, nejčastěji do formátu PCM.
  - 3 Dále jsou převedena zvolenou metodou krátkodobé analýzy na posloupnost vektorů příznaků.
  - 4 Detekce hranic slov
    - může být náročné na provedení např. kvůli rušivému pozadí.
    - Nekorektní detekce hranic slov zhoršuje úspěšnost rozpoznávání
    - Metody odstraňující i jen částečně vliv pozadí zvyšují výpočetní náročnost.
  - 5 Vytvoření referenčních obrazů slov.

# DTW – Metody vytváření referenčních obrazů slov

- Přímé využití obrazů trénovací množiny jako referenčních obrazů slov
  - namluvená slova od jednoho nebo více řečníků jsou použita jako referenční vzory
  - DTW nevyžaduje, aby obrazy téhož slova byly stejně dlouhé, ale z důvodu možnosti aplikace pomocných kritérií je vhodné provést časovou normalizaci každého obrazu.
- Vytváření průměrného vzorového obrazu pro každou třídu slov  $w$ :
  - používají se metody lineárního a nebo dynamického průměrování
  - lineární průměrování:
    - provedeme lineární časovou normalizaci všech akustických obrazů trénovací množiny
    - výsledné referenční složky obrazu určíme jako průměr odpovídajících složek obrazů pro dané slovo
  - dynamické průměrování:
    - vzorový obraz se vytváří použitím algoritmu DTW

# DTW – Metody vytváření referenčních obrazů slov

## pokračování

- Vytváření vzorových obrazů shlukováním
  - Rozdělíme vzorové obrazy pro dané slovo do shluků tak, že obrazy uvnitř shluku jsou si „podobné“ a obrazy z různých shluků jsou „nepodobné“.
  - Shlukování lze realizovat:
    - interaktivně (poloautomaticky) – metoda řetězové mapy, algoritmus ISODATA (viz Levinson, Rabiner , Sondhi – Interactive Clustering Techniques for Selecting Speaker-Independend Reference Templates for Isolated Word Recognition, IEEE Transactions on ASSP, 27, 1979, č 2)
    - automaticky – algoritmy založené na MacQueenově algoritmu (viz např. Komunikace s počítačem mluvenou řečí).

# DTW – praktická realizace klasifikátoru slov

## Rozpoznávání (klasifikace)

- Během klasifikace probíhá zpracování řečového signálu stejně jako při učení:
  - pokud jsou referenční obrazy normalizovány je nutné normalizovat i rozpoznávaná slova.
- Pravidla využívaná při klasifikaci:
  - minimální vzdálenost
  - varianty pravidla nejbližšího souseda

# Redukce výpočetních a paměťových nároků při použití DTW

- Nevýhody DTW:
  - Vysoké paměťové a výpočetní nároky – mohou znesnadňovat klasifikaci v reálném čase i při relativně malém slovníku.
- Metody řešení:
  - hrubá síla – využití drahých paralelních procesorů případně zákaznických obvodů
  - vhodné zakódování parametrů jednotlivých mikrosegmentů referenčních i testovacích obrazů
  - redukce počtu mikrosegmentů akustického obrazu slova – využívají se oblasti spektrální stacionarity řečového signálu
  - snížení výpočetní náročnosti při hledání nejbližšího souseda ve slovníku
    - vhodná volba prohledávacích postupů.



# Redukce výpočetních a paměťových nároků při použití DTW 2.

- Redukce oblasti prohledávání funkce DTW
  - pomocí heuristik do operací porovnávání obrazů.
- Vhodné zakódování parametrů mikrosegmentů
  - využívá vektorovou kvantizaci a kódovou knihu
  - kódová kniha
    - abeceda konečného počtu kvantizovaných vzorků:
    - každý vektor ve vzorku lze nahradit jeho pořadovým číslem
    - při předem definované kódové knize lze dopředu spočítat matici vzájemných vzdáleností mezi kvantizačními vzory.
- Využití oblastí spektrální stacionarity řečového signálu
  - využívá se přítomnost oblastí spektrální stacionarity
  - metoda spektrální stopy:
    - spektrální stopa – spojnice koncových bodů vektorů příznaků
    - aproximace – např. lineárními úseky.

# Redukce výpočetních a paměťových nároků při použití DTW 3.

- Zavedení účinných způsobů vyhledávání nejbližšího souseda.
  - Viz metody prohledávání metrických prostorů.
    - Nutno ověřit, že vzdálenost použitá v algoritmu DTW je metrika.
- Redukce výpočetních nároků pomocí heuristik při porovnávání:
  - Vícestupňový rozhodovací postup:
    - 1 Porovnáváme promluvu proti celému slovníku pomocí pouze několika příznaků.
    - 2 Výstupem je soubor perspektivních kandidátů (řádově jednotky desítek), ve kterém se vyhledává pomocí klasického DTW.
  - Práh zamítnutí:
    - Po každém kroku porovnáváme spočítanou vzdálenost.
    - Překročíme-li experimentálně získanou hodnotu prahu obraz je zamítnut.

# Metoda HMM (Hidden Markov Model)

## Úvod

- Modelování řeči pomocí HMM vychází z následující představy o tvorbě řeči:
  - Hlasové ústrojí se v krátkém čase nachází v jedné z konečně mnoha artikulačních konfigurací – generuje hlasový signál.
  - Přejde do následující konfigurace.
- Tuto činnost lze chápat statisticky.
- Kvantizací akustických vektorů (vytvořením kódové knihy) lze dosáhnout konečnosti všech parametrů odpovídajícího modelu.

# Princip použití HMM pro rozpoznávání

- Jsou generovány dvě vzájemně svázané časové posloupnosti náhodných proměnných:
  - Podpůrný Markovův řetězec – posloupnost konečného počtu stavů.
  - Řetězec konečného počtu spektrálních vzorů.
- Náhodné funkce ohodnocující pravděpodobnostmi vztah vzorů k jednotlivým stavům.
- Pro rozpoznávání řeči nejčastější využívané levo-pravé Markovovy modely.
  - Vhodné pro modelování procesů spjatých se vzrůstajícím časem.

# Markovův proces

- Markovův proces  $G$  se skrytým Markovovým modelem je pětice  $G = (Q, V, N, M, \pi)$ 
  - $Q = \{q_1, \dots, q_k\}$  – množina stavů
  - $V = \{v_1, \dots, v_m\}$  – množina výstupních symbolů
  - $N = (n_{i,j})$  – matice přechodu
    - určuje pravděpodobnost přechodu ze stavu  $q_i$  v čase  $t$  do stavu  $q_j$  v čase  $t_1$
  - $M = (m_{i,j})$  – matice přechodu, určující pravděpodobnost generování akustického vektoru  $v_j$ , v kterémkoliv čase ve stavu  $q_i$
  - $\pi = (\pi_i)$  – vektor pravděpodobností počátečního stavu (pravděpodobnost toho, že  $i$ . stav je počáteční)
- Trojice  $\lambda = (N, M, \pi)$  – soubor parametrů HMM; vytváří model řečového segmentu (slova, ...)
  - např. Vintsjukův model pro slovo:
    - počet stavů 40 — 50 – odvozeno od průměrného počtu mikrosegmentů ve slově (délka mikrosegmentu 10 ms)

# Určení pravděpodobnosti promluvy

- Značíme  $P(O|\lambda)$ .
- Promluva  $O$  standardně zpracována do posloupnosti  $O = (o_1, \dots, o_T)$ .
  - $T$  – počet mikrosegmentů promluvy
  - $o_i$  – odpovídají výstupním symbolům
- Určení  $P(O|\lambda)$  – metoda využívající rekurzivní výpočet odpředu nebo odzadu generované posloupnosti (forward-backward algorithm).

# Určení pravděpodobnosti

## Pokračování

- Výpočet odpředu:
  - $\alpha_i$  – pravděpodobnost přechodu do stavu  $q_i$  při generování posloupnosti  $\{o_1, \dots, o_t\}$  ( $\alpha_i = P(o_1, o_2 \dots o_t, q_i(t) | \lambda)$ )
  - Rekurzivní výpočet:

- 1 inicializace

$$\alpha_1(i) = \pi_i m_i(o_1)$$

pro  $1 \leq i \leq N$

- 2 rekurze pro  $t=1, 2, \dots, T-1$

$$\alpha_{t+1}(j) = \left[ \sum_{i=1}^N \alpha_t(i) n_{i,j} \right] m_j(o_{t+1})$$

pro  $1 \leq j \leq N$ ,  $m(o_t)$  je ekvivalentní zápisu  $m_i(l)$ , pokud  $o_t = v_l$

- 3 Výsledná pravděpodobnost:

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i)$$

# Alternativní způsob výpočtu $P(O|\lambda)$

- Nevýhoda předchozího postupu:
  - ve výsledném vztahu jsou zahrnuty pravděpodobnosti všech možných posloupností stavů délky  $T$
- Lze nahradit výpočtem maximálně pravděpodobné posloupnosti  $Q$ .
- Výpočet realizován pomocí Viterbiova algoritmu:
  - problém řešen rekurzivně s použitím techniky dynamického programování



# Trénování parametrů modelu $\lambda = (N, M, \pi)$

- Nutno stanovit postup při trénování parametrů modelu
  - maximalizace pravděpodobnosti  $P(O|\lambda)$
  - neexistuje analytická metoda k zajištění globálního maxima
  - používají se iterativní algoritmy zajišťující aspoň lokální maximalitu
- Nejpoužívanější postup – Bauman-Welchův algoritmus.
- Problém při trénování modelu:
  - vliv konečné trénovací množiny – čím menší je trénovací množina a čím větší matice  $M$ , tím větší pravděpodobnost, že některé prvky matice budou nastaveny na 0 – problém chybějících (neadekvátních) dat

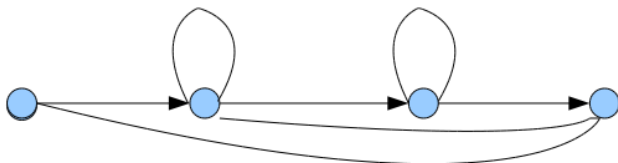
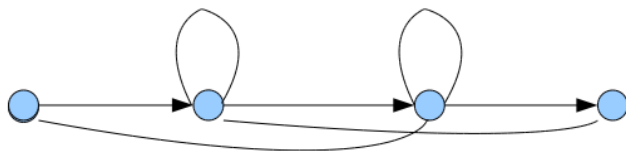
# Rozhodovací pravidlo – rozpoznávání slova

- Princip maximální věrohodnosti:
  - Pro neznámé slovo  $O$  určíme hodnoty  $P(O|\lambda)$  pro všechna modely  $\lambda$ .
  - Jako výsledek vybereme třídu s maximální hodnotou.

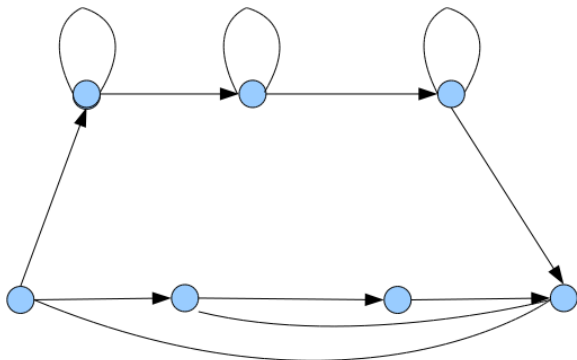
# Implementace

- Modelování povelů:
  - nejčastější se používají modely se 4-7 stavů
  - lze využít SW nástroje pro tvorbu HMM:
    - HTK – Hidden Markov Model ToolKit  
(<http://htk.eng.cam.ac.uk/>)
- Modelování fonémů:
  - obvykle 4-7 stavů
  - model slova – zřetězení modelů fonémů
  - problémy s výpočtem v reálném čase
  - speciální algoritmy na vyhledávání

## Příklady struktur HMM pro fonémy



## Příklady struktur HMM pro fonémy



# Obtíže při rozpoznávání izolovaných slov

- Určení začátku a konce promluvy:
  - šum kontra sykavky
  - detekce nahodilého zvukového vzruchu (klepnutí, ...) kontra okluzívy, které obsahují pauzy
  - možná přítomnost infrazvuků.

# Ukázka plozivy

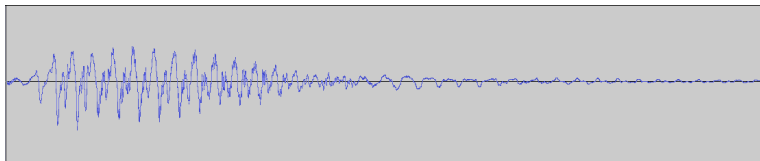


Figure: Hláška P

# Úvod

- Hlavní rozdíly oproti rozpoznávání slov:
  - nelze vytvořit analogii databáze vzorů
  - prozodické faktory
  - nutnost určovat hranice mezi slovy
  - výplňkové zvuky a chyby řeči
- Řešení - statistický přístup
  - použití jazykových modelů
  - HMM vrátí stejnou pravděpodobnost např. pro slova " máma" a " nána"
    - ① máma je častější - vhodné použít máma



# Jazykové modely

- Posloupnost slov (promluva)  $W = (w_1 w_2 \dots w_n)$ .
- Posloupnost akustických vektorů -  $O = O(o_1 o_2 \dots o_t)$ .
- Chceme nalézt  $W^*$  (množinu všech promluv) maximalizující  $P(W|O)$ .
- Dle Bayesova pravidla  
platí:  $P(W^*|O) = \max(P(W|O)) = \max(P(W) * \frac{P(O|W)}{P(O)})$
- Pro nalezení maxima potřebujeme znát:
  - model řečníka  $P(O|W)$
  - jazykový model  $P(W)$
- Model řečníka se nahrazuje pravděpodobností generování  $W$  odpovídajícím Markovovým modelem.
- Trigramový model:
  - Platí:  $P(w_n|w_1..w_{n-1}) \cong P(w_n|w_{n-2}w_{n-1})$

# Rozpoznávání tématu - topic recognition

- Úspěšnost rozpoznávání plynulé řeči 50-99
  - úkolu
  - jazyku
  - mluvčím
  - ...
- Úspěšnost rozpoznávání může zvýšit:
  - znalost tématu promluvy
  - použití gramatiky pro rozpoznávání řeči.
- Mění se stavový prostor a pravděpodobnosti trigramů
  - např. mějme burzovní zprávy - bylo rozpoznáno slovo honey nebo money?
- Známé téma - může být přesnější jazykový model.

# Gramatiky pro podporu rozpoznávání řeči

- Umožňují omezit množinu rozpoznávaných promluv:
  - výhoda - vyšší úspěšnost rozpoznávání
  - nevýhoda - nižší volnost vyjadřování
- Používají se bezkontextové gramatiky.
- V praxi často používané formáty gramatik:
  - JSGF (<http://www.w3.org/TR/jsgf/>) - původně definována v Java Speech API (<http://java.sun.com/products/java-media/speech/>)
  - SRGS (<http://www.w3.org/TR/speech-grammar/>) - součást standardů W3C Voice Browser Activity (<http://www.w3.org/Voice>)
    - Určeny pro tvorbu dialogových a hlasových rozhraní.

# Ukázka gramatiky ve formátu JSGF

```
#JSGF
```

```
<koren> = Chci jet <cim>.|
```

```
        Chci jet <cim> z <odkud> do <kam>.|
```

```
        Chci jet <cim> z <odkud> do <kam> v <kdy>.;
```

```
<cim> = vlakem | autobusem;
```

```
<odkud> = <czMesto>;
```

```
<kam> = <czMesto>;
```

```
<kdy> = <czCas>;
```

# Ukázka odpovídající gramatiky v XML formátu SRGS

```
<grammar root="koren" version="1.0" xml:lang="cs-CZ">
  <rule id="koren">
    <one-of>
      <item>Chci jet <ruleref uri="\#cim"/>.</item>
      <item>Chci jet <ruleref uri="\#cim"/>
        z <ruleref uri="url db názvů stanic"/>
        do <ruleref uri="url db názvů stanic"/>
      </item>
      ...
    </one-of>
  </rule>
```

# Ukázka odpovídající gramatiky v XML formátu SRGS

## Pokračování

```
<rule id="cim">
  <one-of>
    <item tag="vlak">vlakem</item>
    <item tag="autobus">autobusem</item>
    ...
  </one-of>
</rule>
</grammar>
```

# Ukázka gramatiky v ABNF formátu SRGS

```
root=$koren;  
language = cs-CZ;  
...  
$koren = Chci jet $cim. |  
         Chci jet $cim z $<url db stanic>  
         do $<url db stanic>|  
         ...  
$cim = autobusem {$out=autobus} | vlakem {$out=vlak}
```