Počítačové zpracování řeči

Přednáška 6

DTW - dokončení

Dynamické příznaky, alternativní vzdálenost, alternativní DTW, robustnost

Výsledky předchozí úlohy

Cílem bylo porovnat výsledky rozpoznávání (DTW) se 3 typy příznaků na datech 30 osob

Závěry:

- Nejvyšší úspěšnost dosahují kepstrální příznaky (MFCC 12)
- Nejlepší dosažený výsledek pro MFCC: 97.0 %, 2 studenti se dostali nad 90 %
- Nejrychlejší implementace: kompletní experiment s MFCC trval cca 10 s (za tu dobu rozpoznáno 30x40 = 1200 slov)

Vaše poznatky a připomínky:

Dynamické příznaky

Příznaky dosud zmíněné byly vždy určovány v *jednotlivých* framech (bez ohledu na okolní framy). Nazývají se proto **statické**.

Dynamické příznaky – často také nazývané **delta příznaky** – charakterizují *změny* mezi stejnými příznaky v sousedních framech.

Nejjednodušší definice delta příznaků:

$$\Delta f_p(frame) = f_p(frame+1) - f_p(frame-1) \quad ..... \quad (1.\text{derivace})$$

Např. delta energie $\Delta E(frame) = E(frame+1) - E(frame-1)$

speciální případ pro první a poslední frame $\Delta f_p(1) = f_p(2) - f_p(1) \quad \Delta f_p(J) = f_p(J) - f_p(J-1)$

Poznámky:

1) **Dynamické příznaky** občas vedou na **o něco lepší výsledky rozpoznávání** než statické. (Statické hodnoty značně odrážejí vliv hlasitosti řeči, zesílení zvukové karty a mikrofonu, ….), dynamické zase spíše změnu, trendy, atd.
2) Obvykle se však používá **kombinace statických i dynamických příznaků** v rámci jednoho příznakového vektoru. Ten má pak dvojnásobnou dimenzi.
3) V moderních systémech se používají i **delta-delta** příznaky (2. derivace). Počítají se z delta příznaků podle stejných vzorců jako výše.

Problém Euklidovské vzdálenosti

Problém: Různé příznaky mají různý dynamický rozsah.

Např. E může mít hodnoty v rozsahu 5 – 15,
ZCR 10 – 200,
ΔE -1.5 – 1.5

Euklidovská vzdálenost $$d(x_i, r_i) = \sqrt{\sum_{p=1}^{P} (x_{ip} - r_{ip})^2}$$

Při používání Euklidovské vzdálenosti příznaky s **větším rozsahem** mají **výrazně větší vliv** na výslednou hodnotu lokální vzdálenosti. A naopak příznaky jako např. ΔE, budou mít vždy téměř zanedbatelný dopad (byť mohou být významné z hlediska rozpoznávání).

Poznámka:

Roli hraje skutečně **rozsah** a ne vlastní **hodnoty** příznaků. Např. v situaci, kdy

příznak x je v rozsahu 998 - 1002,
příznak y 3 - 25

bude hrát určující roli příznak y. (Je to dáno tím, že euklid. vzdálenost vyhodnocuje rozdíl hodnot.)

Mahalanobisova vzdálenost

Řešení předchozího problému:

Abychom eliminovali vliv rozsahu různých příznaků, měli bychom použít takový postup, který dá všem příznakům **stejnou váhu** při výpočtu vzdálenosti. Jedním z nich je:

Mahalanobisova vzdálenost:

Vychází z toho, že nejlepším měřítkem rozsahu hodnot je statistická veličina **rozptyl**. Čím větší rozptyl, tím menší váhu by měl mít příznak.

$$d(x_i, r_i) = \sqrt{\sum_{p=1}^{P} (x_{ip} - r_{ip})^2 / \sigma_p^2} \qquad \text{kde } \sigma_p^2 \text{ je rozptyl } p\text{-teho priznaku}$$

Poznámky:

1) Rozptyly každého příznaku nejlépe určíme tak, že použijeme veškerá trénovací (referenční) data - po odstranění ticha - a z nich vypočteme rozptyly podle klasického vzorce: $\sigma_p^2 = \frac{1}{K} \sum_{\text{for all K frames}} (x_{kp} - \bar{x}_p)^2$

2) Protože výpočet vzdáleností je nejčastěji opakovanou operací při DTW, výpočet Mahal. vzdál. podle vzorce značně zpomalí rozpoznávání. Efektivnější je předem vydělit hodnoty příznaků **všech slov** (testovaných i ref.) odmocninou z jejich rozptylu (tedy hodnotou σ) a pak již použít klasickou Eukl. vzd.

$$d(x_i, r_i) = \sqrt{\sum_{p=1}^{P} (x_{ip} - r_{ip})^2 / \sigma_p^2} = \sqrt{\sum_{p=1}^{P} \left(\frac{x_{ip}}{\sigma_p} - \frac{r_{ip}}{\sigma_p}\right)^2}$$

Kepstrální vzdálenost

U nejčastěji používaných příznaků typu MFCC se většinou používá zjednodušená vzdálenost – varianta Euklid. vzd. bez odmocniny

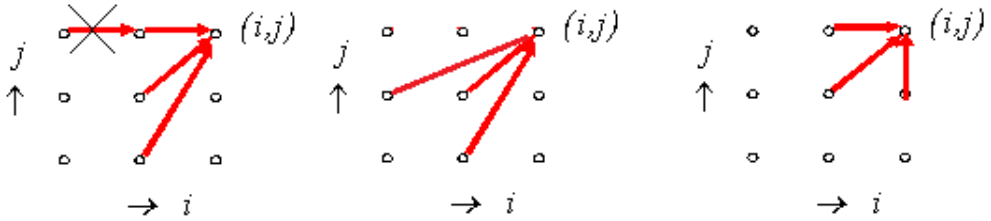
$$d(x_i, r_i) = \sum_{p=1}^{P} (x_{ip} - r_{ip})^2$$

Důvody: kepstrální příznaky mají rozptyl v podobném rozsahu, pro určování míry vzdálenosti není tedy třeba řešit rozptyl a proto ani odstranění odmocniny také nehraje tak velkou roli

Přínosy: zrychlení výpočtu

Alternativní DTW podmínky (1)

V literatuře lze nalézt i jiné definice podmínek spojitosti.



Itakura *varianta1* *var.2 - nejobecnější podmínky*

I pro alternativní podmínky můžeme použít **stejný algoritmus** popsaný v přednášce 4. Změní se pouze vzorec pro akumulovanou vzdálenost A(i,j).

Pro alternativní podmínky dle varianty 2:

$$A(i,j) = d(x_i, r_j) + Min[A(i-1,j), A(i-1,j-1), A(i,j-1)]$$

Poznámky:

1) **Výhody** nejobecnějších alternativních podmínek:
Jsou **symetrické**. Mohou pomoci v úlohách, kde lze přepokládat **velké rozdíly** v trvání promluv.

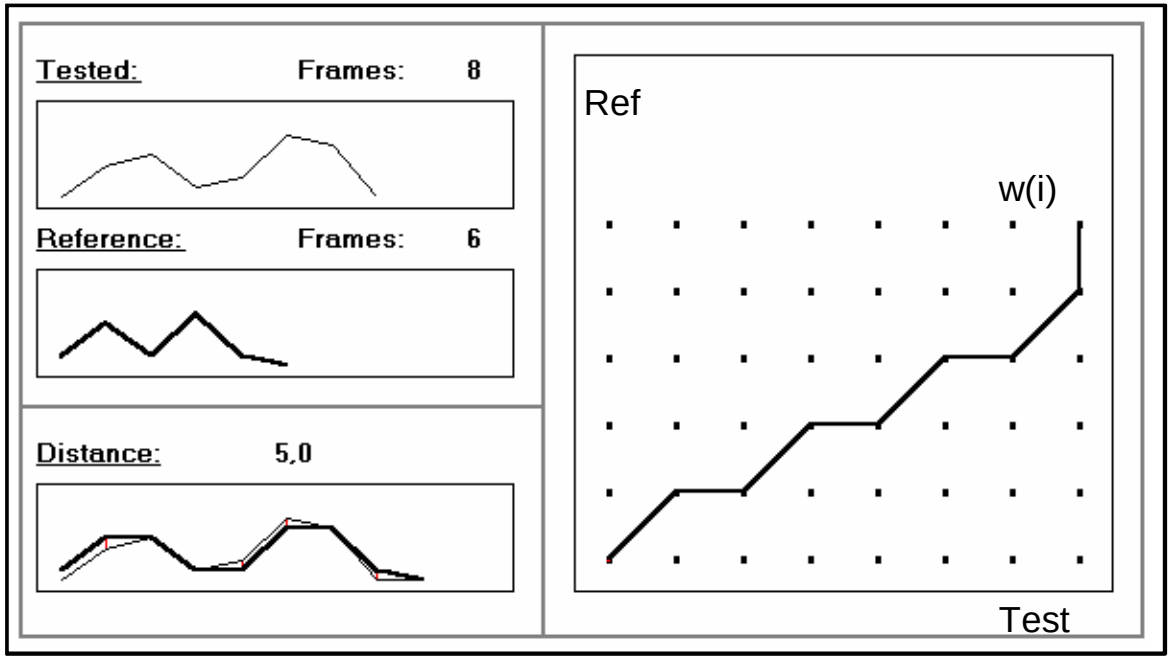
2) **Nevýhody:**
a) Transformační cesta **není funkce**. (Více ref. framů může být přiřazeno k jednomu framu.) Výsledná globální vzdálenost musí být **normalizována**, protože cesta může mít **různou délku** pro různé reference. Normalizace se provádí např. vydělením celkové vzdálenosti hodnotou (I+J).
b) Není zde žádné apriorní globální omezení pro cestu. V praxi lze ale prohledávaný prostor omezit vhodnou globální podmínkou, např. pás podél spojnice bodů (1,1) a (I, J) o určité šířce.

Alternativní DTW podmínky (2)

Příklad a ilustrace:

x = (1, 4, 5, 2, 3, 7, 6, 1) I = 8

r = (2, 5, 2, 6, 2, 1) J = 6



Robustnost rozpoznávání

DTW rozpoznávač je značně závislý na referenční sadě. Ta se skládá z dat od konkrétních osob, a proto vždy funguje lépe pro tyto konkrétní osoby. Nejlépe funguje v tzv. speaker dependent (SD) režimu, kdy rozpoznává řeč osoby na základě referencí od téže osoby.

Rovněž platí, že funguje dobře pouze, jsou-li akustické podmínky během nahrávání referencí stejné jako při rozpoznávání.

Jak udělat rozpoznávač robustnější?

1) Pro **SD** systém: Nahrajte a použijte *více referencí* od téže osoby.
2) Pro **multi-speaker** systém: Nahrajte a využijte reference od *více mluvčích*. **Rada pro praktické nasazení:** Mějte **separátní reference pro mužské a ženské uživatele** a vyberte mezi nimi podle pohlaví daného uživatele.
3) Pro **další zvýšení robustnosti:** Přidejte reference nahrané za různých podmínek
4) Ale … Čím více referencí, tím **rychlejší implementaci** budete potřebovat.
5) **Speaker-independent (SI)** systém se můžete pokusit realizovat jako multi-speaker systém s referencemi od velkého počtu mluvčích (obecně různých od uživatelů).

Vliv detektoru začátku a konce

Úspěšnost DTW rozpoznávače je silně závislá na správném **nalezení začátku a konce slova** (jak u referenčních tak testovacích dat)

Jak tento vliv optimalizovat?

a) **Najít optimální hodnotu prahu.** Lze to udělat automaticky. Na trénovacích datech uskutečnit sérii experimentů s různými hodnotami prahu a vybrat tu, která vede na odložených (vývojových) datech k nejlepším výsledkům. Následně ověřit na datech testovacích.

b) **Nevěřit detektoru absolutně**, „brát ho s rezervou“. Např. takto: Detektorem nalezené hranice posunout o –N framů nazpět (začátek), resp. +N framů dále (konec). Vhodná volba N~ 5-20. Takto vytvořit reference i testovací data. DTW algoritmus už si poradí s optimálním přiřazením skutečného začátku a konce slova. (Jediná nevýhoda velkého N je nárůst výpočetního času u DTW).

Aplikace DTW v jiných oblastech (1)

Princip DTW je použitelný i v dalších oblastech, zejména pro **porovnávání různě dlouhých sekvencí:**

1. **Měření míry podobnosti mezi psanými slovy, např. při kontrole pravopisu: úloha najít nejbližší slovo ke slovu mimo slovník (překlep)**

   Příklad:

   'aplkace' –> 'aplikace', 'duplikace', 'epilace', 'plkat ….

   Lze aplikovat princip DTW (s obecnými podmínkami), je pouze nutné stanovit „vzdálenost" písmen,

   např. když písmena jsou zaměněna, vynechána nebo vložena
   locDist = 1, jinak locDist = 0

   Jiný příklad (chybějící diakritika)
   'Ricany' –> 'Říčany', 'Řežany', 'Míčany' …..

   modifikace: vzdál .mezi stejnými písmeny s/bez diakritiky: locDist = 0.5, jinak 1

Metoda se nazývá Minimum editting distance (MED) nebo Levensteinova vzdálenost

Aplikace DTW v jiných oblastech (2)

2. Podobnost slovních sekvencí :

Např. při vyhodnocování úspěšnosti systémů rozpoznávání spojité řeči:

Příklad:

Text:	Nepřišel jsem	protože		jsem měl		včera příliš	hodně	práce	(9 slov)
ASR:	Ne přišel sem	proto že		jsem měla		včera přílišně		práce	(10 slov)
	I	S	S	S	I	S		S	D

Řešení:

Algoritmem typu MED k sobě přiřadíme obě nestejně dlouhé sekvence a najdeme počet Substitucí (S), Vynechání (D – delece) a Vložení (I – inzerce)

3. Detekce plagiátů – stejný princip jako výše

4. Měření podobnosti sekvencí DNA, apod.

Samostatná úloha

Úprava stávajícího rozpoznávacího systému pro SD a SI experimenty.

- **Jako příznaky budete používat MFCC 12**
- **Na trénovacích datech (s vyloučením testovacích) si zoptimalizujte detektor**
- **Na dodaných datech si zoptimalizujte volbu referenčních sad pro SD a SI experimenty**

Testovací a trénovací sady

Testovací sada společná od teď pro všechny experimenty:
- osoby 21xx a 22xx sady 04-05, tj. 16 osob x 10 slov x 2 sady = 320 slov
- kromě výsledků každé osoby je určující hlavně výsledek přes všechny

Trénovací sady pro SD (speaker dependent) experimenty:
- osoby 21xx a 22xx sady 01-03, v rámci experimentů zjistěte, jak závisí úspěšnost na počtu sad v trénovacím souboru, udělejte to pro případy, kdy trénovací sadu tvoří postupně sada 01, sady 01+02, sady 01+02+03
- protože jde o SD experimenty, tak **pro každou testovanou osobu tvoří trénovací sadu pouze data od této osoby**

Trénovací sady pro SI (speaker independent) experimenty:
- osoby 30 až 49, sady 01-05, tedy 20 osob x 10 slov x 5 = **1000 slov**
- v rámci experimentů zjistěte, jak závisí úspěšnost na počtu sad v trénovacím souboru, udělejte to pro případy, kdy trénovací sadu tvoří postupně sada 01, sady 01+02, sady 01+02+03, … 01+02+03+04+05
- protože jde o SI experimenty, tak pro každou testovanou osobu tvoří trénovací sadu data od všech osob trénovacího souboru

Experimenty SD a SI vyhodnocujte zvlášť. Do pondělí prosím vaše výsledky.