

Dynamické borcení času DTW

Jan Černocký, FIT VUT Brno

April 6, 2005

V tomto cvičení si ukážeme aplikaci dynamického borcení času (DTW – Dynamic Time Warping) v rozpoznávání izolovaných slov. Pomocí DTW srovnáváme dvě posloupnosti vektorů: referenční $\mathbf{R} = [\mathbf{r}(1), \dots, \mathbf{r}(R)]$ o délce R a testovací $\mathbf{O} = [\mathbf{o}(1), \dots, \mathbf{o}(T)]$ o délce T . Chceme znát jejich vzdálenost $D(\mathbf{O}, \mathbf{R})$. V nejjednodušším případě odpovídá i -tému slovu w_i ve slovníku jen jedna reference \mathbf{R}_i , rozpoznané slovo je pak dáno jako:

$$i^* = \arg \min_i D(\mathbf{O}, \mathbf{R}_i). \quad (1)$$

Při DTW definujeme matici \mathbf{D} o rozměrech $T \times R$, kterou zaplníme lokálními vzdálenostmi vektorů: $d(i, j) = d(\mathbf{o}(i), \mathbf{r}(j))$. Vzdálenost $d(\cdot, \cdot)$ je například cepstrální míra. Druhou důležitou maticí je \mathbf{G} – matice částečných kumulovaných vzdáleností. Oproti \mathbf{D} má \mathbf{G} navíc nultý řádek a nultý sloupec, které inicializujeme hodnotami ∞ , kromě $g(0, 0) = 0$. Pro lokální omezení cesty typu I. a pro váhy typu a. (viz přednáška o DTW) můžeme další prvky \mathbf{G} vypočítat:

$$g(i, j) = \min \begin{cases} g(i-1, j) + d(i, j), \\ g(i, j-1) + d(i, j), \\ g(i-1, j-1) + 2d(i, j) \end{cases} \quad (2)$$

pro $1 \leq i \leq T$ a $1 \leq j \leq R$. Poslední prvek této matice $g(T, R)$ udává celkovou optimální vzdálenost:

$$D(\mathbf{O}, \mathbf{R}) = \frac{g(T, R)}{T + R} \quad (3)$$

V matici \mathbf{G} mohu pak zpětně vyhledat optimální cestu, podle které se sekvence srovnávaly. Všechny tyto operace jsou implementovány ve funkci `dtw.m` v balíku `demo.tgz`.

1 Úkol 1. – DTW v Matlabu

V balíku `demo.tgz` máte nachystané demo, které se předvádí na přednášce. Rozbalte je a podívejte se do souboru `readme.txt`, jak je pustit.

Jako parametrizace jsou použity LPC-cepstrální koeficienty (viz přednáška o LPC). Ve funkci `c_matrice.m` poněkud podivně implementován výpočet LPC-koeficientů (stačilo by na každý rámeček zavolat funkci `lpc`, zřejmě jsem si před lety, kdy jsem `c_matrice` psal, potřeboval dokázat, že tomu rozumím...). Podívejte se ale do funkce `a_to_cepst.m` na implementaci převodu LPC koeficientů na LPCC — v přednášce je to úplně na konci, rovnice (38).

Vlastní implementace DTW je ve funkci `dtw.m` - má relativně obsáhlý help, podívejte se i dovnitř. Při volání funkce `dtw` si všimněte jejího grafického výstupu: první panel zobrazuje matici lokálních vzdáleností \mathbf{D} . Druhý panel matici částečných kumulovaných vzdáleností \mathbf{G} . Třetí panel zobrazuje optimální cestu.

2 Úkol 2. – DTW v C

Vášim úkolem bude napsat (či spíše dopsat) jednoduchý rozpoznávač DTW v C. V balíku `reco_dtw.tgz` naleznete soubory potřebné pro kompilaci dvou exáčů:

2.1 fea_lpcc

Počítá ze signálů LPC-cepstrální koeficienty a ukládá je do souboru.

Vášim úkolem je použít Vaše funkce pro výpočet LPC koeficientů (měli byste mít 100% funkční výpočet autokorelačních koeficientů a Levinsona-Durbina) a jen přidat volání pro konverzní funkci `LpcToCps`, která je pro Vás připravena v souborech `lpcc.[ch]`. Pozor, podívejte se důkladně na její help – nechce nultý LPC-koeficient, takže pokud máte vektor \mathbf{A} definovaný včetně nultého, je potřeba ji zavolat

```
LpcToCps (A+1, params, LPC_COEFFS, LPCC_COEFFS);
```

Soubor s parametry tentokrát doplníme na začátku jedním `int`-em s počtem vektorů (ušetří nám to pak problémy při alokaci paměti). V mainu `fea_lpcc.c` je toto už udělané.

Vypočítaný soubor parametrů laskavě srovnajte s tím, co jste dostali v Matlabu, nejlépe na jednom ze souborů z `dema`:

Prikaz radka:

```
reco_dtw/fea_lpcc demo/1.116 fea/1.cps
```

Matlab:

```
sigceps                                % toto je kus dema v Matlabu
ff = fopen ('fea/1.cps','r');
N = fread (ff,[1 1],'int')
c1c = fread (ff,[10 inf],'float');
fclose (ff);

c1c - c1
```

...měli byste vidět velmi malé rozdíly ($\times 10^{-4}$).

2.2 reco_dtw

je vlastní rozpoznávač — je hotový, a není třeba v něm nic dopisovat Volá se jako:

```
reco_dtw N ref1.cps ref2.cps ... refN.cps test1.cps test2.cps ...
```

kde N je počet referencí, následují soubory s referencemi a pak libovolný počet testovacích souborů na rozpoznání.

- Proveďte LPCC parametrizaci několika referencí a několika testovacích souborů. Můžete se inspirovat skriptem `fea_lpcc_all.sh`.
- Spusťte rozpoznávání na souborech z dema. Ověřte, zda `reco_dtw` dává stejné DTW vzdálenosti a rozpoznává stejná slova jako Matlab. Pokud byly LPCC-matice stejné, měl by.
- Prostudujte, jak je rozpoznávač udělán, zaměřte se zvláště na funkci `DTWDistance` z `dtw.c`. Pokuste se najít vztah s přednáškou o DTW, není to příliš složité.
- Vyzkoušejte si rozpoznávání na vlastních referencích - namluvte si několik řečových souborů (např. finalistů SuperStar), pak pár testovacích a ověřte, zda rozpoznávač funguje. Pak požádejte o namluvení testů přítele/přítelkyni. Je DTW-rozpoznávač speaker-independent? Jak byste nezávislosti na řečníkovi dosáhli?