

Skryté Markovovy modely (HTK)

Jan Černocký, FIT VUT Brno

Toto cvičení se zabývá trénováním a rozpoznáváním pomocí HMM. Tentokrát ovšem nebudeme používat Matlab, ale toolkit pro HTK (Hidden Markov model toolkit) z University v Cambridge (UK).

1 HTK

slouží k definici, trénování a k rozpoznávání pomocí HMM a dále obsahuje pomůcky pro parametrizaci řečových signálů (feature extraction), vyhodnocování výsledků rozpoznávání, práci s výslovnostními slovníky, atd. HTK je napsáno v jazyce C a pro nekomerční použití je možné jej stáhnout z

<http://htk.eng.cam.ac.uk/>

My budeme využívat sadu předkompilovaných programů, které budeme spouštět pod operačním systémem Windows NT. Použité programy budou:

- **HCopy** - jak název praví, měl by kopírovat. Při kopírování ovšem dokáže udělat i libovolnou konverzi dat, např. ze signálu na MFCC koeficienty. Chování specifikujeme pomocí konfiguračního souboru.
- **HList** - zobrazuje soubor s parametry.
- **HCompV** - inicialisuje parametry “vysílacích” hustot rozdělení pravděpodobnosti ve stavech modelu na globální hodnoty pro dané slovo.
- **HRest** - přetrénování modelu – provádí výpočet “měkké” funkce přiřazení jednotlivých vektorů ke stavům (state occupation likelihood), s následným přetrénováním parametrů.
- **HParse** - konvertuje lidsky čitelnou formu rozpoznávací sítě do nečitelné HTK podoby.
- **HVite** - Viterbiho dekodér neboli rozpoznávač. Pro natrénované modely a neznámá slova provede výpočet Viterbiho pravděpodobností a výběr maxima. Model, který “vyslal” slovo s největší pravděpodobností, je rozpoznán.
- **HResults** - pomůcka pro vyhodnocení výsledků rozpoznání – na základě správných přepisů rozpoznávaných slov vyhodnocuje chybovost.

Po spuštění libovolného programu bez parametrů se objeví nápověda.

2 Zadání

Vytořte rozpoznávač pro izolovaná slova ANO/NE nezávislý na mluvčím. K natrénování použijte data od 60 mluvčích z databáze “Číslovky” (každý řekl ano i ne). K testování použijte data od 20 mluvčích.

Parametrizaci proveďte pomocí 12-ti MFCC koeficientů a logaritmu energie (v HTK notaci MFCC_E). Vektor doplňte odhady prvních a druhých derivací (Δ a $\Delta\Delta$ koeficienty, v HTK notaci MFCC_E_D_A). Šířka okna pro výpočet jednoho rámce je 25 ms, posuv okna (frame shift) je 10 ms, dostanete tedy 100 vektorů za 1 sekundu.

Modely budou levo-pravé, se zakázaným přeskočením stavu. Z i -tého stavu je tedy možný přechod jen zase do i -tého nebo do $i + 1$ -ího. Modely budou mít celkem 7 stavů, první a poslední jsou speciální nevysílací, vysílacích stavů bude tedy 5. Funkce hustoty rozdělení pravděpodobnosti v jednotlivých stavech budou modelovány pomocí jedné Gaussovy funkce s diagonální kovarianční maticí. Jedna funkce hustoty rozdělení pravděpodobnosti bude tedy popsána vektorem 39-ti středních hodnot a vektorem 39-ti rozptylů (variance).

3 Řešení a komentáře

Vzhledem k typu úlohy je zde obsaženo celé řešení. V případě zájmu se prosím podívejte na soubor README¹, který obsahuje informace o seznámech pro HTK, výrobě MLF souborů, prototypů HMM atd.

¹Mnohé příkazy z README Vám ovšem poběží pouze pod operačním systémem UNIX.

Praktické poznámky

Do libovolného adresáře na disku D:, C: nebo N: si nakopírujte obsah L:\ASS\HTK. Obsahy jednotlivých adresářů:

- `cfg` — konfigurační soubory pro HTK programy.
- `dics` — slovníky.
- `net` — slovní sítě pro rozpoznávání.
- `lists` — seznamy modelů.
- `proto` — prototypy modelů.
- `hmm0` — modely inicializované pomocí HCompV.
- `hmm1` — modely přetrénované pomocí HRest.
- `data` — řečová data v obvyklém formátu: bez hlavičky, $F_s=8000$ Hz, 16-bit lin. Do stejných adresářů se budou generovat i soubory s MFCC koeficienty. Soubory `*a0.raw` obsahují ANO, soubory `*a1.raw` obsahují NE.
- `mlf` — popisy řečových dat v Master-Label souborech.
- `scripts` — seznamy souborů pro HTK. Název je zavádějící, protože “skripty” se obvykle označují sekvence příkazů pro operační systém (např. `.bat` v DOSu). Bohužel je toto označení i v dokumentaci k HTK...

Otevřete si okno s příkazovou řádkou (zde budete spouštět HTK programy) a Windows Commander: pomocí F3 můžete prohlížet obsah textových souborů, pomocí F4 modifikovat.

3.1 Parametrisace

- Prostudujte konfigurační soubor `cfg\hcopy.conf`:

<code>BYTEORDER</code>	<code>= VAX</code>	pořadí byte bude Intel-PC
<code>SOURCEKIND</code>	<code>= WAVEFORM</code>	
<code>SOURCEFORMAT</code>	<code>= NOHEAD</code>	řečové soubory jsou bez hlavičky
<code>SOURCERATE</code>	<code>= 1250</code>	jejich vzorkovací perioda je 1250×100 ns
<code>ZMEANSOURCE</code>	<code>= FALSE</code>	nebudeme ustředňovat signály
<code>TARGETKIND</code>	<code>= MFCC_E_D_A</code>	typ výstupních parametrů: MFCC a log-energie, delty a double-delty.
<code>TARGETFORMAT</code>	<code>= HTK</code>	
<code>TARGETRATE</code>	<code>= 100000</code>	vzorkovací perioda výstupních vektorů bude 10 ms
<code>WINDOWSIZE</code>	<code>= 250000.0</code>	a délka okna 25 ms
<code>NUMCHANS</code>	<code>= 24</code>	počet trojúhelníkových filtrů pro výpočet MFCC
<code>ENORMALISE</code>	<code>= TRUE</code>	normování energie.
- Prohlédněte si skripty `scripts\train.scf` a `scripts\test.scf`.
- Spusťte parametrisaci pro trénovací i testovací set:

```
HCcopy -T 1 -C cfg\hcopy.conf -S scripts\train.scf
HCcopy -T 1 -C cfg\hcopy.conf -S scripts\test.scf
```
- Prohlédněte si některý z vytvořených souborů jako text (pomocí programu HList) nebo v Matlabu pomocí funkce `readhtk.m`.

4 Trénování

4.1 Inicialisace modelů

- Prohlédněte si prototypy modelů v adresáři `proto`. Všimněte si, že v matici přechodových pravděpodobností na konci modelu jsou “zadrátovány” povolené a zakázané přechody. Střední hodnoty jsou nastaveny na nuly, rozptyly na jedničky.

- Prohlédněte si Master-Label soubor `mlf\train.mlf`. Čísla před značkou udávají začátek a konec souboru opět ve stovkách ns. Zkontrolujte pro jeden soubor, zda délka zaznamenaná v MLF “sedí”: počet byte / 2 / 8000 / 100×10^{-9} .
- Prohlédněte si skript `scripts\train_htk.scp`
- Spusťte inicialisaci pro oba modely:
`HCompV -T 7 -I mlf\train.mlf -l ANO -m -S scripts\train_htk.scp -M hmm0 proto\ANO`
`HCompV -T 7 -I mlf\train.mlf -l NE -m -S scripts\train_htk.scp -M hmm0 proto\NE`
- Prohlédněte si modely v adresáři `hmm0`. Co vidíte ?

4.2 Přetrénování modelů

- Spusťte přetrénování pro oba modely:
`HRest -T 7 -I mlf\train.mlf -l ANO -S scripts\train_htk.scp -M hmm1 hmm0\ANO`
`HRest -T 7 -I mlf\train.mlf -l NE -S scripts\train_htk.scp -M hmm1 hmm0\NE`
- Prohlédněte si modely v adresáři `hmm1`. Co vidíte ?

5 Rozpoznávání a vyhodnocení

5.1 Co budeme ještě potřebovat

Z trénování máme v adresáři `hmm1` k dispozici dva natrénované modely. K rozpoznávání ovšem budeme ještě potřebovat:

- seznam modelů. Viz soubor. `lists\models`.
- výslovnostní slovník (pronunciation dictionary). Tento slovník udává, jaké modely budou použity pro které slovo v případě, kdy pro rozpoznávání používáme modely menších jednotek (fonémů), např. ANO=A N O. V našem případě máme na jedno slovo jeden model, výslovnostní slovník je tedy triviální: `dics\dictionary`.
- síť pro rozpoznávání (recognition network). Síť udává, jaké kombinace slov se mohou objevit na výstupu rozpoznávače. Pro nás to může být pouze ANO nebo NE. Ručně vytvořená a lidsky čitelná síť je v `net\oldnetwork`. Tu je třeba překonvertovat do podoby, kterou umí přečíst už jen HTK:
`HParse net\oldnetwork net\network`

5.2 Rozpoznávání

vyprodukuje pro neznámé soubory přepis a uloží jej opět do MLF: `mlf\testout.mlf`. Rozpoznávání spustíme pomocí:

```
HVite -T 1 -d hmm1 -S scripts\test_htk.scp \
-i mlf\testout.mlf -w net\network dics\dictionary lists\models
```

- podívejte se na výsledky rozpoznání ve výstupním Master Label souboru.

5.3 Vyhodnocení

Velice by nás zajímalo, jak rozpoznávání dopadlo. Máme k dispozici MLF se správným přepisem testovacích souborů: `mlf\test.mlf`. Tento správný přepis můžeme srovnat s výstupem HVite pomocí:

```
HResults -I mlf\test.mlf lists\models mlf\testout.mlf
```

Nejdůležitějším číslem na výstupu je `Acc=` (word accuracy), která udává úspěšnost rozpoznání slov. Kolika % jste dosáhli ?

6 A teď sami ! Pokud stiháte...

1. Nahrajte si 10 vlastních souborů s ANO. Pro HCopy použijte konfigurační soubor `hcopy_wav.conf`, který dokáže pracovat se soubory WAV. Vytvořte vlastní skripty pro parametrizaci a pro rozpoznávání. Vytvořte vlastní MLF pro tyto soubory (pokud MLF použijete pouze jako referenci pro HResults, není nutné zadávat časy). Parametrizujte pomocí HCopy a rozpoznajte pomocí HVite. Vyhodnoťte chybovost pomocí HResults. Kolika % jste dosáhli ?

2. Zašumte v Matlabu Vaše soubory bílým šumem tak, aby poměr signálu k šumu SNR byl 0 dB. Původní soubory např. přesuňte do `xx_clean.wav` a pak použijte následující sekvenci příkazů:

```
SNR = 0;
[s,fs,nbit] = wavread('xx_clean.wav');
s = s' - mean(s);
E = sum(s.^2) / length(s);
Enoise = E / 10^(SNR/10);
n = randn(1,length(s)) * sqrt(Enoise);
wavwrite (s + n, fs, nbit,'xx.wav');
```

Výsledné signály si **poslechněte**, ať víte s čím má rozpoznávač “tu čest” (uvědomte si, že pro SNR=0 dB je energie signálu stejná jako energie šumu). Proveďte MFCC parametrizaci, rozpoznajte a vyhodnoťte chybovost. Změnila se ?