Corso di Teorie e Tecniche del Riconoscimento A.A. 2012/2013

Esercizi Matlab

Marco Cristani

18 dicembre 2012

1 Hidden Markov Models

Scompattate lo zip della lezione in una directory locale. I file che vedete sono codici MATLAB necessari all'addestramento e all'inferenza di modelli HMM (All'interno della directory troverete dei file che non verranno utilizzati nella lezione corrente, ma che possono tornarvi utili per una trattazione più estesa ed approfondita dei modelli).

1.1 Generazione di dati, modelli di Markov

Dato il modello Markoviano $\tilde{\boldsymbol{\lambda}}_1 = \{\tilde{\mathbf{A}}_1, \tilde{\pi}_1\}$ così caratterizzato:

$$\tilde{\mathbf{A}}_1 = \begin{bmatrix} a_{1,1} = 0.05 & a_{1,2} = 0.9 & a_{1,3} = 0.05 \\ a_{2,1} = 0.9 & a_{2,2} = 0.1 & a_{2,3} = 0 \\ a_{3,1} = 0.5 & a_{3,2} = 0.5 & a_{3,3} = 0 \end{bmatrix} \tag{1}$$

$$\tilde{\pi}_1 = \begin{bmatrix} \pi_1 = 0.3 & \pi_2 = 0.3 & \pi_3 = 0.4 \end{bmatrix} \tag{2}$$

generate 2 sequenze lunghe 1000 campioni, $\mathbf{S}_{1,1}$ e $\mathbf{S}_{1,2}$, chiamando gli stati con i seguenti simboli:

- $S_1 = 10$
- $S_2 = 20$
- $S_3 = 30$

I vettori $\mathbf{S}_{1,1}$ e $\mathbf{S}_{1,2}$ sono delle sequenze di stati (che devono essere di dimensioni 1×1000). A questo punto simulate un processo d'osservazione "sporcando" con del rumore gaussiano di deviazione standard $\sigma = 2$ i vettori $\mathbf{S}_{1,1}$ e $\mathbf{S}_{1,2}$. Così facendo otterrete i vettori d'osservazione $\mathbf{O}_{1,1}$ e $\mathbf{O}_{1,2}$ di uguale dimensionalità rispetto ai vettori di stato.
Fate lo stesso prendendo in considerazione il processo Markoviano $\tilde{\boldsymbol{\lambda}}_2 = \{\tilde{\mathbf{A}}_2, \tilde{\pi}_2\}$ così caratterizzato:

$$\tilde{\mathbf{A}}_2 = \begin{bmatrix} a_{1,1} = 0.5 & a_{1,2} = 0.3 & a_{1,3} = 0.1 \\ a_{2,1} = 0.5 & a_{2,2} = 0.1 & a_{2,3} = 0.1 \\ a_{3,1} = 0 & a_{3,2} = 0.6 & a_{3,3} = 0.8 \end{bmatrix} \tag{3}$$

$$\tilde{\pi}_2 = \left[\ \pi_1 = 0.8 \quad \pi_2 = 0.1 \quad \pi_3 = 0.1 \ \right] \tag{4}$$

ottenendo il vettore di stati $\mathbf{S}_{2,1}$ ed il vettore d'osservazione $\mathbf{O}_{2,1}$.
Visualizzate i tre vettori, evidenziando i primi 30 campioni.

1.2 Addestramento Hidden Markov Model

A questo punto, considerando il vettore $\mathbf{O}_{1,1}$, addestrate una HMM $\boldsymbol{\lambda}_1 = \{\mathbf{A}_1, \pi_1, \mathbf{b}_1\}$ tramite i seguenti comandi:

```
load foptions_DATA;     % specifica i criteri di fermata di Baum Welch;
Hmm_1                   =   init_mhmm(O_11,K,Q,'diag',left_right,foptions_DATA);
[LL,Hmm_1,nit]          =   learn_ghmm(O_11,Hmm_1,max_iter)
```

dove K = 3 è il numero di stati del processo, $Q = 1$ indica che l'osservazione degli stati è Gaussiana monodimensionale, left_right = 0 specifica un particolare tipo di HMM che qui non ci interessa e max_iter = 100 è il numero massimo di iterazioni per il processo di training di Baum Welch. Una volta terminato l'addestramento, valutate i parametri ottenuti, traendo le opportune considerazioni.

1.3 Calcolo di likelihood tramite Hidden Markov Model

Valutate le (log)likelihood

- $P(\mathbf{O}_{1,1}|\boldsymbol{\lambda}_1)$
- $P(\mathbf{O}_{1,2}|\boldsymbol{\lambda}_1)$
- $P(\mathbf{O}_{2,1}|\boldsymbol{\lambda}_1)$

tramite il seguente codice

```
XX             =    mk_ghmm_obs_lik(osservazione, Hmm_1.m, Hmm_1.s);
[alpha,xi,mloglik[]] =   forwards(Hmm_1.pi, Hmm_1.A, XX);
```

avendo cura di rinominare la variabile XX al fine di differenziare le varie inferenze. Traetene le opportune considerazioni.

1.4 Calcolo di cammini di massima probabilitá tramite Hidden Markov Model

Provate la seguente funzione

```
path = viterbi_path(Hmm_1.pi, Hmm_1.A, XX);
```

Funzioni utili di Matlab

Oltre alle funzioni elencate sopra, è utile anche la seguente:

Estrazione da distribuzioni discrete: *extract_disc_state*