

University of Surrey, Guildford GU2 7XH. University of Surrey, Guildford GU2 7XH.

HMM tutorial 2

by Dr Philip Jackson

- Recap. of MMs and HMMs
- Computing likelihoods
 - Markov model
 - Hidden Markov model
- Finding best state sequence
 - Viterbi algorithm
 - Trellis diagram
- Summary



from (Young et al. 1997)



Recapitulation: fundamentals

Maximum likelihood estimation



Bayesian estimation



Markov Model, ${\cal M}$

(a) Initial-state probabilities,

$$\pi = \{\pi_i\} = \{P(x_1 = i)\}$$
 for $1 \le i \le N$;

(b) State-transition probabilities,

$$A = \{a_{ij}\} = \{P(x_t = j | x_{t-1} = i)\} \text{ for } 1 \le i, j \le N;$$



producing a sequence $X = \{1, 1, 2, 3, 3, 4\}.$

Hidden Markov Model, λ

- (a) Initial-state probabilities, $\pi = \{\pi_i\} = \{P(x_1 = i)\}$ for $1 \le i \le N$;
- (b) State-transition probabilities, $A = \{a_{ij}\} = \{P(x_t = j | x_{t-1} = i)\} \text{ fo}$

for
$$1 \leq i, j \leq N$$
;

(c) Discrete output probabilities,

$$B = \{b_i(k)\} = \{P(o_t = k | x_t = i)\}$$

for
$$1 \le i \le N$$

and $1 \le k \le K$.



producing observations with a state sequence $X = \{1, 1, 2, 3, 3, 4\}.$

Three problems for HMMs

- 1. Compute likelihoods $P(\mathcal{O}|\lambda)$;
- 2. Find best state sequence X^* ;
- 3. Re-estimate model parameters $\Lambda = \{\lambda\}$.

Likelihood for MM state sequence



Transition probabilities:

$$\pi = \{\pi_i\} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \text{ and}$$
$$A = \{a_{ij}\} = \begin{bmatrix} 0.8 & 0.2 \\ 0 & 0.6 \end{bmatrix}.$$

Probability of a certain state sequence, $X = \{1, 2, 2\}$:

$$P(X|\mathcal{M}) = \pi_1 a_{12} a_{22}$$
$$= 1 \cdot$$
$$=$$

(21)

Likelihood for HMM state sequence



Output probabilities:
$$B = \begin{bmatrix} b_1(k) \\ b_2(k) \end{bmatrix} = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0 & 0.9 & 0.1 \end{bmatrix}$$
.

Probability with a certain state sequence, $X = \{1, 2, 2\}$:

$$P(\mathcal{O}, X|\lambda) = P(\mathcal{O}|X, \lambda)P(X|\lambda)$$

= $\pi_1 b_1(o_1) a_{12} b_2(o_2) a_{22} b_2(o_3)$
= $1 \cdot$
 \approx

(22)

Problem 1: Computing $P(\mathcal{O}|\lambda)$

Joint probability of the observations and state sequence, for a given model λ :

$$P(\mathcal{O}, X|\lambda) = P(\mathcal{O}|X, \lambda) P(X|\lambda) = \pi_1 b_1(o_1) a_{11} b_1(o_2) a_{12} b_2(o_3) \dots (23)$$

To get the total probability of the observations, we must sum across all possible state sequences:

$$P(\mathcal{O}|\lambda) = \sum_{X} P(\mathcal{O}|X,\lambda)P(X|\lambda)$$
$$= \sum_{X} P(\mathcal{O},X|\lambda).$$
(24)

Forward procedure

Consider
$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, x_t = i | \lambda)$$
:

1. Initially, $\alpha_1(i) = \pi_i b_i(o_1),$ for $1 \le i \le N;$

2. For
$$t = 2, 3, ..., T$$
,
 $\alpha_t(j) = \left[\sum_{i=1}^N \alpha_{t-1}(i) a_{ij}\right] b_j(o_t), \quad \text{for } 1 \le j \le N;$
(25)

3. Finally,

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

Thus, we can solve Problem 1 efficiently by recursion.

Worked example of the forward procedure



Backward procedure

Define
$$\beta_t(i) = P(o_{t+1}, o_{t+2}, ..., o_T | x_t = i, \lambda)$$
:

1. Initially, $\beta_T(i) = 1$, for $1 \le i \le N$;

2. For
$$t = T - 1, T - 2, ..., 1$$
,
 $\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$, for $1 \le i \le N$;
(26)

3. Finally,

$$P(\mathcal{O}|\lambda) = \sum_{i=1}^{N} \pi_i b_i(o_1) \beta_1(i).$$

We now have another efficient way of computing $P(\mathcal{O}|\lambda)$.

Worked example of the backward procedure



Problem 2: best state sequence

Given observations $\mathcal{O} = \{o_1, \dots, o_T\}$, find the state sequence $X = \{x_1, \dots, x_T\}$ with greatest likelihood:

$$X^* = \arg \max_{X} P(\mathcal{O}, X | \lambda)$$

= $\arg \max_{X} \Delta(X)$ (27)

where

$$\Delta(X) = \pi_{x_1} b_{x_1}(o_1) \prod_{t=2}^T a_{x_{t-1}x_t} b_{x_t}(o_t). \quad (28)$$

The Viterbi algorithm is an inductive algorithm that allows us to find the optimal state sequence X^* efficiently.

Step 1



1. Initially,

$$\begin{split} \delta_1(i) &= \pi_i b_i(o_1) \\ \psi_1(i) &= 0 \end{split} \qquad \qquad \text{for } 1 \leq i \leq N; \end{split}$$

Step 2



2. For
$$t = 2, ..., T$$
,
 $\delta_t(j) = \max_i \left[\delta_{t-1}(i) a_{ij} \right] b_j(o_t)$
 $\psi_t(j) = \arg \max_i \left[\delta_{t-1}(i) a_{ij} \right]$ for $1 \le j \le N$;

Steps 3 and 4



3. Finally,

$$\Delta^* = \max_i [\delta_T(i)]$$

$$x_T^* = \arg \max_i [\delta_T(i)];$$

4. Trace back, for t = T - 1, T - 2, ..., 1, $x_t^* = \psi_{t+1} \left(x_{t+1}^* \right)$, and $X^* = \{ x_1^*, x_2^*, ..., x_T^* \}$. (29)

Viterbi by numbers



Reformulating the optimisation

Recall the likelihood calculation,

$$P(\mathcal{O}, X|\lambda) = P(\mathcal{O}|X, \lambda)P(X|\lambda)$$

= $\pi_1 b_1(o_1) a_{11} b_1(o_2) a_{12} b_2(o_3) \dots$

Now, taking the negative logarithm of eq. 28 gives

$$Q(X) = -\left[\ln\left(\pi_{x_1} b_{x_1}(o_1)\right) + \sum_{t=2}^T \ln\left(a_{x_{t-1}x_t} b_{x_t}(o_t)\right)\right].$$
(30)

Hence, eq. 27 becomes

$$X^* = \arg\min_X Q(X). \tag{31}$$

Summary of the Viterbi algorithm

1. Initially,

$$q_1(i) = -\ln \pi_i - \ln (b_i(o_1))$$

 $\psi_1(i) = 0$ for $1 \le i \le N$;

2. For
$$t = 2, ..., T$$
,
 $q_t(j) = \min_i \left[q_{t-1}(i) - \ln a_{ij} \right] - \ln \left(b_j(o_t) \right)$
 $\psi_t(j) = \arg \min_i \left[q_{t-1}(i) - \ln a_{ij} \right]$ for $1 \le j \le N$;

- 3. Finally, $Q^* = \min_i [q_T(i)]$ $x_T^* = \arg\min_i [q_T(i)];$
- 4. Trace back, for $t = T 1, T 2, \dots, 1$, $x_t^* = \psi_{t+1} \left(x_{t+1}^* \right)$, and $X^* = \{ x_1^*, x_2^*, \dots, x_T^* \}$. (32)

Tutorial 2 summary

- Recap. of MMs and HMMs
- Computing likelihoods, $P(\mathcal{O}|\lambda)$
 - for Markov models
 - for Hidden Markov models
- \bullet Finding the best state sequence, X^{\ast}
 - Viterbi algorithm
 - Trellis diagrams

Next

- Setting the parameters in the models $\Lambda = \{\lambda\}$
 - Baum-Welch re-estimation
 - Forward-backward algorithm
 - Continuous output pdfs

Further reading

L. R. Rabiner. A tutorial on HMM and selected applications in speech recognition. In Proc. IEEE, Vol. 77, No. 2, pp. 257–286, 1989.