ICT600 Computational Mathematics

Hidden Markov Models

Composed by Cholwich Nattee

Disclaimer: this partial note is my attempt to help you learn the material in this course. Things in here might not be in the real exam and vice versa. Don't use this as your main study. There might be some typos and/or mistakes.

Be advised: these partial notes have been created for the sole purpose of aiding your studies in this class, and are *for personal use only*. They may not be duplicated, copied, modified or translated without my written consent.

.-**-..-**-..-**-..-**-..-**-.∼ ♡ Have Fun!♡ ∽.-**-..-**-..-**-..-**-..-**-.

1 Markov Chains

Definition 1. A *Markov chain* is a sequence of variables $X_1, \dots, X_t$ where the probability distribution of X_t depends on only a *finite fixed number* of previous variables.

Definition 2. A *first-order Markov chain* is a Markov chain where the probability distribution X_t depends on only the value of X_{t-1}, i.e.

$$\mathbf{P}\left(X_t \mid X_1, X_2, \dots, X_{t-1}\right) = \mathbf{P}\left(X_t \mid X_{t-1}\right)$$

Definition 3. A *finite first-order Markov chain* is defined as a 3-tuple as:

(1) a finite set of states $Q = \{1, 2, \dots, k\}$

(2) transition probabilities between two states $A = \{a_{ij}\}$ where

$$a_{ij} = p(X_t = j | X_{t-1} = i)$$

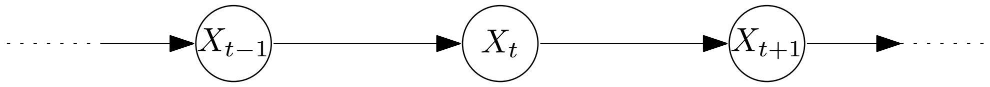
with a constraint that $\sum_{j=1}^{k} a_{ij} = 1$ for all $i \in Q$,

(3) initial state probabilities $\pi = \{\pi_i\}$ where

$$\pi_i = p(X_0 = i)$$

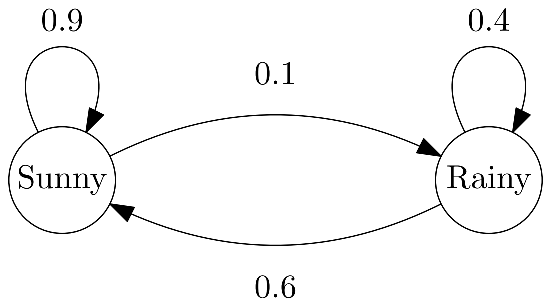
with $\sum_{i=1}^{k} a_{0i} = 1$

The graphical model of a Markov chain is shown below.



Example 1. In one city, a sunny day is 90% likely to be followed by another sunny day. A rainy day is 40% likely to be followed by another rainy day. The weather of day 0 is known to be “sunny”. Write the formal representation of the Markov chain corresponding to this situation.

Each Markov chain can be depicted as a *state diagram*. Each state represents a state. Each transition between states is attached with a probability value.



Example 2. Find the probability that the weather of day 2 is "rainy".

2 Hidden Markov Model

Definition 4. A *Hidden Markov Model* (HMM) is a finite Markov chain where its state at time t cannot be directly observed. Each state however emits a symbol with a certain probability. Each HMM is defined as a 5-tuple:

(1) a finite set of states $Q = \{1, 2, \dots, k\}$

(2) a finite set of symbols $\Sigma = \{b_1, b_2 \dots, b_m\}$

(3) transition probabilities $A = \{a_{ij}\}$ where

$$a_{ij} = p(X_t = j \mid X_{t-1} = i)$$

with $\sum_{j=1}^{k} a_{ij} = 1$ for all $i \in Q$,

(4) initial state probabilities $\pi = \{\pi_i\}$

$$\pi_i = p(X_0 = i)$$

with $\sum_{i=1}^{k} \pi_i = 1$

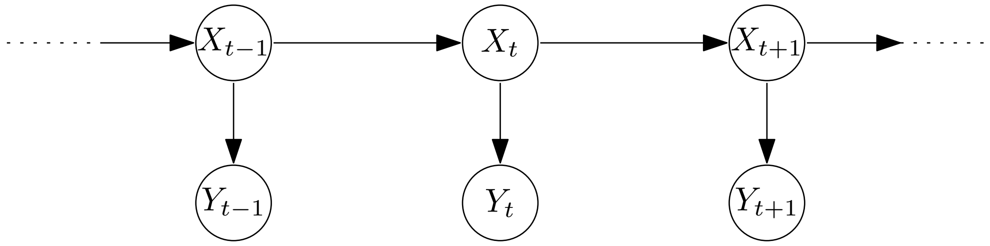
(5) Emission probabilities $E = \{e_i(b)\}$

$$e_i(b) = p(Y_t = b \mid X_t = i)$$

with $\sum_{b \in \Sigma} e_i(b) = 1$ for all $i \in Q$,

Similar to the Markov chain, an HMM can be depicted as a state diagram. Every state transition is labeled with a transition probability. Every state is attached with the emission probabilities.

The graphical model of an HMM is shown below.



Example 3. A casino has two dice:

(1) Fair die: $p(1) = p(2) = p(3) = p(4) = p(5) = p(6) = \frac{1}{6}$, and

(2) Loaded die: $p(1) = p(2) = p(3) = p(4) = p(5) = \frac{1}{10}$; $p(6) = \frac{1}{2}$

The dealer may switch between the dice with the probability that the dealer switches from the fair die to the loaded die is 0.01, and the probability that the dealer switches from the loaded die to the fair die is 0.2. However, the dealer always starts with the fair die. Write the formal definition and draw the diagram for the HMM.

Example 4. Let λ_2 be an HMM defined as follow:

(1) States: $Q = \{1, 2, 3\}$

(2) Observation symbols: $\Sigma = \{a, b, c, d\}$

(3) Transition probabilities

$$\begin{array}{llllll} a_{11} = 0.1 & a_{12} = 0.4 & a_{13} = 0.5 & a_{21} = 0.5 & a_{22} = 0.2 & a_{23} = 0.3 \\ a_{31} = 0.2 & a_{32} = 0.6 & a_{33} = 0.2 & & & \end{array}$$

(4) Emission probabilities

$$\begin{array}{llll} e_1(a) = 0.05 & e_1(b) = 0.25 & e_1(c) = 0.4 & e_1(d) = 0.3 \\ e_2(a) = 0.25 & e_2(b) = 0.25 & e_2(c) = 0.25 & e_2(d) = 0.25 \\ e_3(a) = 0.5 & e_3(b) = 0.1 & e_3(c) = 0.3 & e_3(d) = 0.1 \end{array}$$

(5) Initial state probabilities

$$a_{01} = 1.0 \qquad a_{02} = 0.0 \qquad a_{03} = 0.0$$

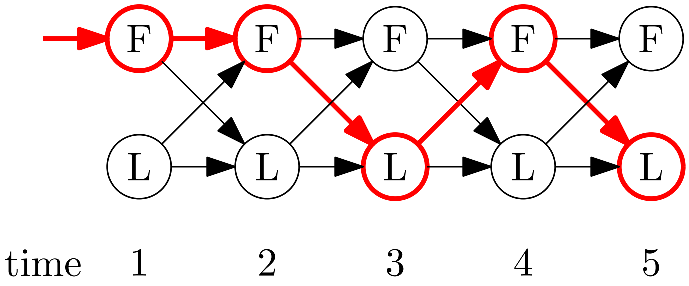
Draw a state diagram showing this HMM λ_2.

2.1 Parse and Trellis

Definition 5. A *parse* is a sequence of visited states at each time.

Each parse is typically depicted by a *trellis* which is a directed graph. Each node in the trellis represents a state at one time. Each edge in the trellis shows a state transition from one time to another.

Example 5. From the HMM for the fair and loaded dice in the previous example, we can draw a trellis for a parse Fair → Fair → Loaded → Fair → Loaded as below.



2.2 HMM Inference

2.2.1 Evaluation

Evaluation is a task to compute the probability that an observation sequence is generated given an HMM.

Likelihood of a Parse Given an HMM λ, a parse $X = \langle x_1, \ldots, x_n \rangle$, and an observation sequence $Y = \langle y_1, \ldots, y_n \rangle$, the likelihood of the parse producing the observation sequence (given the HMM λ) can be computed as

$$\begin{aligned}
&p(y_1, \ldots, y_n, x_1, \ldots, x_n \mid \lambda) \\
&\qquad = p(y_1, \ldots, y_n, x_1, \ldots, x_n) \\
&\qquad = p(x_1)p(y_1 \mid x_1) \prod_{t=2}^{n} p(x_t \mid x_{t-1})p(y_t \mid x_t) && (2.1) \\
&\qquad = p(y_1, \ldots, y_{n-1}, x_1, \ldots, x_{n-1})p(y_i|x_i)p(x_{i-1}|x_i) && (2.2)
\end{aligned}$$

Example 6. Let λ_1 be the HMM in the previous example. Given an observation sequence $Y = \langle 1, 1, 6, 6, 6\rangle$ and a parse $X = \langle F, F, L, F, L\rangle = \langle 1, 1, 2, 1, 2\rangle$, compute the likelihood of the parse producing the observation sequence $p(Y, X \mid \lambda_1)$.

Example 7. Compute $p\left(a, b, c, S_1, S_2, S_3 \mid \lambda_2\right)$

Likelihood of a Sequence

Let $Y = \langle y_1, y_2, \ldots, y_n \rangle$ be an observation sequence and λ be an HMM,

$$p(Y \mid \lambda) = p(Y) = \sum_{X \in \mathcal{X}} p(Y, X) \tag{2.3}$$

where $\mathcal{X}$ is a set of all possible parses with length of n.

Example 8. Compute $p(1, 4, 6|\lambda_1)$

Forward algorithm

Computing $p(Y)$ directly from Equation 2.3 is a time-consuming task since many redundant calculations are performed. The dynamic programming approach can be used to speed up the calculation. We define the *forward probability* $f_k(i)$ as the probability of generating the first i symbol in the observation sequence and ending up in state k.

$$\begin{aligned}
f_k(i) &= p(y_1, \dots, y_i, x_i = k) \\
&= \sum_{x_1,\dots,x_{i-1}} p(y_1, \dots, y_i, x_1, \dots, x_{i-1}, x_i = k) \\
&= \sum_{l} \sum_{x_1,\dots,x_{i-2}} p(y_1, \dots, y_i, x_1, \dots, x_{i-2}, x_{i-1} = l, x_i = k) \\
&= \sum_{l} \left[\left[\sum_{x_1,\dots,x_{i-2}} p(y_1, \dots, y_{i-1}, x_1, \dots, x_{i-2}, x_{i-1} = l) \right] a_{lk} e_k(y_i) \right] \\
&= \sum_{l} p(y_1, \dots, y_{i-1}, x_{i-1} = l) a_{lk} e_k(y_i) \\
&= e_k(y_i) \sum_{l} f_l(i-1) a_{lk} \qquad (2.4)
\end{aligned}$$

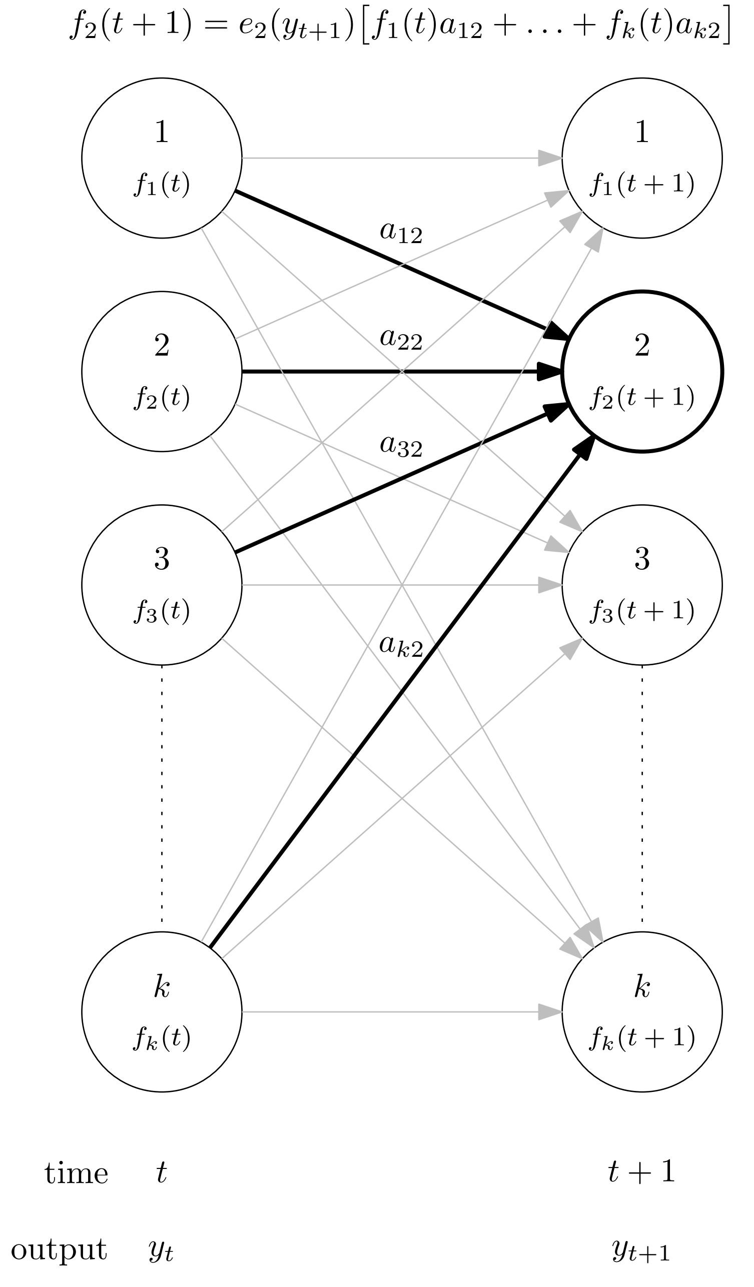
where $f_k(1) = \pi_k e_k(y_1)$

We can then compute $p(Y)$ based on the forward probability as

$$p(Y) = \sum_{k} f_k(n) \qquad (2.5)$$

where n is the length of Y.

The following figure shows how the *forward probability* is computed.



Forward Algorithm

```
k = 2

a = [[0.99, 0.01],
     [0.20, 0.80]]

e = [{’1’:0.166, ’2’:0.166, ’3’:0.166,
      ’4’:0.166, ’5’:0.166, ’6’:0.166},
     {’1’:0.100, ’2’:0.100, ’3’:0.100,
      ’4’:0.100, ’5’:0.100, ’6’:0.500}]

pi = [1.0, 0.0]

Y = ’146’

f = [[0.0 for i in enumerate(Y)] for j in range(k)]

# Initialization
for i in range(k):
    f[i][0] = pi[i]*e[i][Y[0]]

# Induction
for t in range(1, len(Y)):
    for i in range(k):
        s = 0.0
        for l in range(k):
            s += f[l][t-1]*a[l][i]
        f[i][t] = e[i][Y[t]]*s

# Termination
t = len(Y)-1
p = 0.0
for i in range(k):
    p += f[i][t]

print(p)
```

Example 9. Compute $p(1, 4, 6 \mid \lambda_1)$ using the forward algorithm.

Example 10. Compute $p(a, c, a, a \mid \lambda_2)$ using the forward algorithm.

2.3 Decoding

Decoding is a task to find a parse for a given observation sequence that maximizes the likelihood of the parse.

Given an HMM λ and an observation sequence $Y = \langle y_1, \ldots, y_n \rangle$, find

$$X^* = \arg\max_X p(Y, X|\lambda) \tag{2.6}$$

Similar to the forward algorithm, we can use the dynamic programming approach to perform the task. The algorithm is called "*Veterbi algorithm*". We define

$$V_k(i) = e_k(y_i) \max_{l=1,\ldots,k} \left[V_l(i-1)a_{lk}\right] \tag{2.7}$$

where $V_k(1) = \pi_k e_k(y_1)$

Example 11. Find the parse X that maximizes $p(1, 4, 6 \mid \lambda_1)$

Example 12. Find the parse X that maximizes $p(a, c, a, a \mid \lambda_2)$

Veterbi Algorithm

```
k = 2

states = [’Fair’, ’Loaded’]

a = [[0.99, 0.01],
     [0.20, 0.80]]

e = [{’1’:0.166, ’2’:0.166, ’3’:0.166,
      ’4’:0.166, ’5’:0.166, ’6’:0.166},
     {’1’:0.100, ’2’:0.100, ’3’:0.100,
      ’4’:0.100, ’5’:0.100, ’6’:0.500}]

pi = [1.0, 0.0]

Y = ’146’

V = [[0.0 for i in enumerate(Y)] for j in range(k)]

W = [[0 for i in enumerate(Y)] for j in range(k)]

# Initialization
for i in range(k):
    V[i][0] = pi[i]*e[i][Y[0]]

# Induction
for t in range(1, len(Y)):
    for i in range(k):
        v = []
        for l in range(k):
            v.append(V[l][t-1]*a[l][i])

        V[i][t] = e[i][Y[t]]*max(v)
        W[i][t] = v.index(max(v))

# Termination
v = []
t = len(Y)-1
for i in range(k):
    v.append(V[i][t])
vi = v.index(max(v))
X = states[vi]

for i in range(t-1, -1, -1):
    vi = W[vi][i]
    X = states[vi] + ’ ’ + X

print(X)
```

2.4 Learning

Given a training set

$$\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^{N}$$

where each X_i is a parse, and Y_i is an observation sequence generated from X_i.

This *learning* task is to estimate the parameters (A, B, π) that best explains the training set. We can use a technique called "*maximum likelihood estimation (MLE)*" to find the parameters.

$$a_{ij} = p(X_t = j \mid X_{t-1} = i) = \frac{Count(i \rightarrow j)}{Count(i)} \tag{2.8}$$

where $Count(i \rightarrow j)$ returns the number of $i \rightarrow j$ transitions found in the parses in $\mathcal{D}$. Similarly, $Count(i)$ is the number of state i in $\mathcal{D}$.

$$e_k(b) = p(Y_t = b \mid X_t = k) = \frac{Count(b, k)}{Count(k)} \tag{2.9}$$

where $Count(b, k)$ is the number of times that a symbol b is emitted from state k, and $Count(k)$ is the number of state k.

$$\pi_i = p(X_1 = i) = \frac{Count(X_1 = i)}{N} \tag{2.10}$$

where $Count(X_1 = i)$ is the number of parses having i as the first state.

Example 13. From the following examples, estimate the parameters of an HMM λ_3 with $Q = \{1, 2, 3, 4\}$ and $\Sigma = \{a, b, c, d\}$:

Parse	Observation sequence
$1, 2, 1, 3, 4, 4$	a, b, b, a, c, d
$2, 1, 3, 4, 4$	b, b, a, c, d
$1, 1, 1, 2, 2, 4$	c, d, d, c, a, a
$2, 3, 3, 3, 4, 4, 1$	c, a, b, c, d, a, b

References

- Lawrence R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”, Proceedings of the IEEE, Vol. 77, No. 2, February 1989.