Hidden Markov Models-5
Parameter Learning

M. R. Hasan
Readings

• Alpaydin: 15
• Russell & Norvig: 15
What We Will Cover

- Problem: Parameter Learning
- Baum-Welch Algorithm
Background Assumed

- Recursion & Dynamic Programming
- Complexity Theory
- Graphical Models
- d-separation
HMM: Four Inference Problems

- Given a number of sequences of observations, we are interested in four inference problems:

**Problem 1 (Evaluation):** Given a model $\lambda$, we would like to evaluate the probability of any given observation sequence $O = \{O_1 O_2 \ldots O_T\}$: $P(O \mid \lambda)$

As part of the evaluation problem, we solve two sub-problems:

**Filtering:** Compute the posterior distribution over the most recent state given all evidence (observation) to date.

**Prediction:** Compute the posterior distribution over the future state given all evidence (observation) to date.
HMM: Four Inference Problems

**Problem 2 (Smoothing):** Given a model $\lambda$, we would like to compute the posterior distribution over a past state given all evidence up to the present $O = \{O_1 O_2 \ldots O_T\}$: $P(q_t = S_i \mid O_{1:T}, \lambda)$

**Problem 3 (Most likely Explanation):** Given a model $\lambda$ and an observation sequence $O$, we would like to find out the state sequence $Q = \{q_1 q_2 \ldots q_T\}$, which has the highest probability of generating $O$; namely, we want to find $Q^*$ that maximizes $P(Q \mid O, \lambda)$.

**Problem 4 (Learning):** Given a training set of observation sequences, $X = \{O^k\}_{k}$, we would like to learn the model that maximizes the probability of generating $X$; namely, we want to find $\lambda^*$ that maximizes $P(X \mid \lambda)$. 
Problem 1 (Evaluation): Given a model $\lambda$, we would like to evaluate the probability of any given observation sequence $O = \{O_1 O_2 \ldots O_T\}$: $P(O \mid \lambda)$

As part of the evaluation problem, we solve two sub-problems:

Filtering: Compute the posterior distribution over the most recent state given all evidence (observation) to date.

Prediction: Compute the posterior distribution over the future state given all evidence (observation) to date.

Problem 2 (Smoothing): Given a model $\lambda$, we would like to compute the posterior distribution over a past state given all evidence up to the present $O = \{O_1 O_2 \ldots O_T\}$: $P(q_t = S_i \mid O_{1:T}, \lambda)$

Problem 3 (Most likely Explanation): Given a model $\lambda$ and an observation sequence $O$, we would like to find out the state sequence $Q = \{q_1 q_2 \ldots q_T\}$, which has the highest probability of generating $O$; namely, we want to find $Q^*$ that maximizes $P(Q \mid O, \lambda)$.

Problem 4 (Learning): Given a training set of observation sequences, $X = \{O^k\}_k$, we would like to learn the model that maximizes the probability of generating $X$; namely, we want to find $\lambda^*$ that maximizes $P(X \mid \lambda)$. 
Problem 4 (Learning): Given a training set of observation sequences, \( X = \{O^k\}_k \), we would like to learn the model that maximizes the probability of generating \( X \); namely, we want to find \( \lambda^* \) that maximizes \( P(X \mid \lambda) \).

- **Setup for HMM:**
  - \( N \): Number of states in the model: \( S = \{S_1, S_2, \ldots, S_N\} \)
  - \( M \): Number of observation symbols in the alphabet: \( V = \{v_1, v_2, \ldots, v_m\} \)
  - State transition probabilities: \( A = [a_{ij}] \)
    
    Where \( a_{ij} \equiv P(q_{t+1} = S_j \mid q_t = S_i) \)
  - Observation probabilities: \( B = [b_j(m)] \)
    
    Where \( b_j(m) \equiv P(O_t = v_m \mid q_t = S_j) \)
  - Initial state probabilities: \( \Pi = [\pi_i] \)
    
    Where \( \pi_i \equiv P(q_1 = S_i) \)
  - Parameter set of an HMM: \( \lambda = (A, B, \Pi) \)
Problem 4 (Learning): Given a training set of observation sequences, $X = \{O^k\}_k$, we would like to learn the model that maximizes the probability of generating $X$; namely, we want to find $\lambda^*$ that maximizes $P(X | \lambda)$.

- Given an observation sequence $O$ and the dimensions $N$ and $M$, find the model $\lambda = (A, B, \Pi)$ that maximizes the probability of $O$.
- This can be viewed as training a model to best fit the observed data.
- Alternatively, we can view this as a (discrete) hill climb on the parameter space represented by $A, B, \text{and } \Pi$.

N: Number of states in the model: $S = \{S_1, S_2, \ldots, S_N\}$
M: Number of observation symbols in the alphabet: $V = \{v_1, v_2, \ldots, v_m\}$
Problem 4: Learning

• Here we want to **adjust the model parameters to best fit** the observations.

• The sizes of the matrices (N and M) are fixed but the elements of \( \mathbf{A}, \mathbf{B} \) and \( \Pi \) are to be determined, subject to the row stochastic condition (i.e., each row values sum to 1).

• The fact that we can **efficiently re-estimate the model itself** is one of the most amazing aspects of HMMs.

\[
N: \text{Number of states in the model: } S = \{S_1, S_2, \ldots, S_N\}
\]

\[
M: \text{Number of observation symbols in the alphabet: } V = \{v_1, v_2, \ldots, v_m\}
\]
Problem 4: Learning

• Let’s provide the intuition for estimating the transition probability $a_{ij}$ and emission probability $b_j(m)$.
• Given the whole observation $O$, the estimated transition probability $\hat{a}_{ij}$:

\[
\hat{a}_{ij} = \frac{\text{Expected number of Transitions from } S_i \text{ at } t \text{ to } S_j \text{ at } t + 1}{\text{Total number of transitions from } S_i \text{ at } t}
\]
Problem 4: Learning

- To **mathematically represent** the estimated transition probability $\hat{a}_{ij}$, we define two parameters: $\xi_t(i, j)$ & $\gamma_t(i)$

- $\xi_t(i, j)$: the probability of being in $S_i$ at time $t$ and transitioning to state $S_j$ at time $t+1$, given the whole observation $O$ and $\lambda$.

$$\xi_t(i, j) \equiv P(q_t = S_i, q_{t+1} = S_j \mid O, \lambda)$$

- $\gamma_t(i)$: the probability of being in state $S_i$ at time $t$

$\gamma_t(i)$ provides the **total number of transitions** from state $S_i$ at time $t$
Problem 4: Learning

$\gamma_t(i)$: the probability of being in state $S_i$ at time $t$

$\gamma_t(i)$ provides the total number of transitions from state $S_i$ at time $t$

$\gamma_t(i)$ is calculated by marginalizing over the arc of all probabilities for all possible next states ($S_j$ at time $t+1$).

$$\gamma_t(i) \equiv \sum_{j=1}^{N} \xi_t(i, j)$$

$\xi_t(i, j) \equiv P(q_t = S_i, q_{t+1} = S_j | O, \lambda)$
Problem 4: Learning

\[ \gamma_t(i) \equiv \sum_{j=1}^{N} \xi_t(i, j) \]

\[ \xi_t(i, j) \equiv P(q_t = S_i, q_{t+1} = S_j | O, \lambda) \]

Based on the two parameters \( \xi_t(i, j) \) and \( \gamma_t(i) \), we **mathematically represent** the estimated transition probability \( \hat{a}_{ij} \) as follows:

\[ \hat{a}_{ij} = \frac{\text{Expected number of Transitions from } S_i \text{ at } t \text{ to } S_j \text{ at } t + 1}{\text{Total number of transitions from } S_i \text{ at } t} \]

\[ \hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \]
Problem 4: Learning

- The **estimated emission probability** $\hat{b}_j(m)$ is defined as follows.
- $\hat{b}_j(m)$: is the **probability of observing** $\nu_m$ when the system is in state $S_j$

$$
\hat{b}_j(m) = \frac{\text{Expected number of times } \nu_m \text{ is observed when the system is in state } S_j}{\text{Total number of times the system is in state } S_j}
$$

$$
\hat{b}_j(m) = \frac{\sum_{t=1}^{T} \gamma_t(j) 1(O_t = \nu_m)}{\sum_{t=1}^{T} \gamma_t(j)}
$$
\[ h_j(m) = \frac{\text{Expected number of times } v_m \text{ is observed when the system is in state } S_j}{\text{Total number of times the system is in state } S_j} \]

Numerator: consider the following figure.

All hidden states \{S_1, \ldots, S_j, \ldots, S_N\} at time \( t \) generate observations.

We count how many such observations are equal to \( v_m \) (\( O_t = v_m \)).

Then, we divide the observation count of \( v_m \) by the total number of observations.

It is equal to the number of times the system is in state \( S_j \) at time \( t \).
Let us now discuss how to adjust the model parameters to best fit the observations.

In other words, given an observation sequence $O$ and the dimensions $N$ and $M$, find the model $\lambda = (A, B, \Pi)$ that maximizes the probability of $O$.

This is done by iteratively re-estimating the model parameters $\lambda = (A, B, \Pi)$. 
Problem 4: Learning

- The model can be re-estimated as follows.

1. For $i = 1, 2, \ldots, N$, let

   \[ \pi_i = \gamma_1(i) \]

2. For $i = 1, 2, \ldots, N$, and $j = 1, 2, \ldots, N$, compute

   \[ \hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \]

3. For $j = 1, 2, \ldots, N$, and $m = 1, 2, \ldots, M$, compute

   \[ \hat{b}_j(m) = \frac{\sum_{t=1}^{T} \gamma_t(j) 1(O_t = v_m)}{\sum_{t=1}^{T} \gamma_t(j)} \]

This iterative re-estimation is given by the Baum-Welch algorithm.
Problem 4: Learning

- Baum-Welch algorithm is executed in **two steps**.

**Step 1**: Compute the expected values (parameters $\xi_t(i, j)$ and $\gamma_t(i)$) for the model parameters based on the current model parameters $\lambda = (A, B, \Pi)$.

**Step 2**: Re-estimate the model parameters $\lambda = (A, B, \Pi)$ to maximize the probability of the observation sequence $P(O | \lambda)$.

We iterate the EM steps as long as the observation sequence probability $P(O | \lambda)$ increases.

1. For $i = 1, 2, \ldots, N$, let $\pi_i = \gamma_1(i)$

2. For $i = 1, 2, \ldots, N$, and $j = 1, 2, \ldots, N$, compute $\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$

3. For $j = 1, 2, \ldots, N$, and $m = 1, 2, \ldots, M$, compute $b_j(m) = \frac{\sum_{t=1}^{T} \gamma_t(j) \mathbb{1}(O_t = o_m)}{\sum_{t=1}^{T} \gamma_t(j)}$
Problem 4: Learning

- The summary of the iterative steps of the Baum-Welch algorithm is given below.

E-step:

1. Initialize the model $\lambda = (A, B, \Pi)$.
2. Compute $\xi_t(i, j)$ and $\gamma_t(i)$.

M-step:

3. Re-estimate the model $\lambda = (A, B, \Pi)$.
4. If $P(O | \lambda)$ increases, go to step 2 (or quit after a max number of iterations)

\begin{align*}
\pi_i &= \gamma_t(i) \\
\hat{a}_{ij} &= \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \\
\hat{b}_j(m) &= \frac{\sum_{t=1}^{T} \gamma_t(j) \mathbf{1}(O_t = v_m)}{\sum_{t=1}^{T} \gamma_t(j)}
\end{align*}
Problem 4: Learning

• Note on Initialization:
• Initialize the model parameters $\lambda = (A, B, \Pi)$ with a best guess.
• If no reasonable guess is available, choose random values, as follows:

$$\pi_i \approx \frac{1}{N} \quad a_{ij} \approx \frac{1}{N} \quad b_j(m) \approx \frac{1}{M}$$

It’s critically important that $A, B$ and $\Pi$ be randomized.

Otherwise exactly uniform values will result in a local maximum from which the model cannot climb.

Also, ensure that $A, B$ and $\Pi$ are row-stochastic.

E-step:
1. Initialize the model $\lambda = (A, B, \Pi)$.
2. Compute $\xi_t(i, j)$ and $\gamma_t(i)$.

M-step:
3. Re-estimate the model $\lambda = (A, B, \Pi)$.
4. If $P(\theta | \lambda)$ increases, go to step 2 (or quit after a max number of iterations)
Problem 4: Learning

• Now given the summary of the Baum-Welch algorithm, let’s discuss **how to compute** $\xi_t(i, j)$ and $\gamma_t(i)$.

• Recall, for $t = 1, 2, \ldots, T-1$ and $i, j \in \{1, 2, \ldots, N\}$, we define $\xi_t(i, j)$ as the probability of being in $S_i$ at time $t$ and transitioning to state $S_j$ at time $t+1$, given the whole observation $O$ and $\lambda$:

$$\xi_t(i, j) \equiv P(q_t = S_i, q_{t+1} = S_j \mid O, \lambda)$$

$$\xi_t(i, j) = \frac{P(O \mid q_t = S_i, q_{t+1} = S_j, \lambda)P(q_t = S_i, q_{t+1} = S_j \mid \lambda)}{P(O \mid \lambda)}$$

Using the following observation:

$$P(a, b \mid c, d) = \frac{P(c \mid a, b, d)P(a, b \mid d)}{P(c \mid d)}$$
Problem 4: Learning

• Derivation of:

\[
P(a, b \mid c, d) = \frac{P(c \mid a, b, d)P(a, b \mid d)}{P(c \mid d)}
\]

\[
P(a, b, c, d) = P(a, b \mid c, d)P(c, d)
\]
\[
= P(a, b \mid c, d)P(c \mid d)P(d)
\]

\[
P(a, b, c, d) = P(c \mid a, b, d)P(a, b, d)
\]
\[
= P(c \mid a, b, d)P(a, b \mid d)P(d)
\]

\[
P(a, b \mid c, d)P(c \mid d)P(d) = P(c \mid a, b, d)P(a, b \mid d)P(d)
\]

\[
P(a, b \mid c, d)P(c \mid d) = P(c \mid a, b, d)P(a, b \mid d)
\]

\[
P(a, b \mid c, d) = \frac{P(c \mid a, b, d)P(a, b \mid d)}{P(c \mid d)}
\]

\[
\xi_t(i, j) \equiv P(q_t = S_i, q_{t+1} = S_j \mid O, \lambda)
\]

\[
\xi_t(i, j) = \frac{P(O \mid q_t = S_i, q_{t+1} = S_j, \lambda)P(q_t = S_i, q_{t+1} = S_j \mid \lambda)}{P(O \mid \lambda)}
\]
Problem 4: Learning

\[ \xi_t(i, j) = \frac{P(O \mid q_t = S_i, q_{t+1} = S_j, \lambda)P(q_t = S_i, q_{t+1} = S_j \mid \lambda)}{P(O \mid \lambda)} \]

Numerator: we factor the 2\textsuperscript{nd} term using following observation:

\[ P(a, b \mid c) = P(b \mid a, c)P(a \mid c) \]

Justification:

\[ P(a, b, c) = P(a, b \mid c)P(c) \]

\[ P(a, b, c) = P(b \mid a, c)P(a, c) \]

\[ = P(b \mid a, c)P(a \mid c)P(c) \]

\[ P(a, b \mid c)P(c) = P(b \mid a, c)P(a \mid c)P(c) \]

\[ P(a, b \mid c) = P(b \mid a, c)P(a \mid c) \]
Problem 4: Learning

\[ \xi_t(i,j) = \frac{P(O \mid q_t = S_i, q_{t+1} = S_j, \lambda)P(q_{t+1} = S_j \mid q_t = S_i, \lambda)P(q_t = S_i \mid \lambda)}{P(O \mid \lambda)} \]

Observe: \( O = \{O_1 \; O_2 \; \ldots \; O_t \; O_{t+1} \; O_{t+2} \; \ldots \; O_T\} \)

\[ \xi_t(i,j) = \frac{P(O_1 \; O_2 \; \ldots \; O_t \; O_{t+1} \; O_{t+2} \; \ldots \; O_T \mid q_t = S_i, q_{t+1} = S_j, \lambda)P(q_{t+1} = S_j \mid q_t = S_i, \lambda)P(q_t = S_i \mid \lambda)}{P(O \mid \lambda)} \]

Numerator: 1\textsuperscript{st} term is factorized using d-separation

\[ \xi_t(i,j) = \frac{[P(O_1 \; O_2 \; \ldots \; O_t \mid q_t = S_i, \lambda)P(O_{t+1} \mid q_{t+1} = S_j, \lambda)P(O_{t+2} \; \ldots \; O_T \mid q_{t+1} = S_j, \lambda)]a_{ij}P(q_t = S_i \mid \lambda)}{P(O \mid \lambda)} \]
Problem 4: Learning

\[ \xi_t(i,j) = \frac{P(O_1 O_2 \ldots O_t \mid q_t = S_i, \lambda) P(O_{t+1} \mid q_{t+1} = S_j, \lambda) P(O_{t+2} \ldots O_T \mid q_{t+1} = S_j, \lambda)}{P(O \mid \lambda)} a_{ij} P(q_t = S_i \mid \lambda) \]

Numerator: Combine the 1st and the last term

\[ \xi_t(i,j) = \frac{P(O_1 O_2 \ldots O_t \mid q_t = S_i, \lambda) P(q_t = S_i \mid \lambda) P(O_{t+1} \mid q_{t+1} = S_j, \lambda) P(O_{t+2} \ldots O_T \mid q_{t+1} = S_j, \lambda)}{P(O \mid \lambda)} a_{ij} \]

\[ \xi_t(i,j) = \frac{P(O_1 O_2 \ldots O_t, q_t = S_i \mid \lambda) P(O_{t+1} \mid q_{t+1} = S_j, \lambda) P(O_{t+2} \ldots O_T \mid q_{t+1} = S_j, \lambda)}{P(O \mid \lambda)} a_{ij} \]
Problem 4: Learning

\[ \xi_t(i, j) = \frac{P(O_1 O_2 \ldots O_t, q_t = S_i | \lambda) P(O_{t+1} | q_{t+1} = S_j, \lambda) P(O_{t+2} \ldots O_T | q_{t+1} = S_j, \lambda) a_{ij}}{P(O | \lambda)} \]

\[ \xi_t(i, j) = \frac{\alpha_t(i) b_j(O_{t+1}) \beta_{t+1}(j) a_{ij}}{P(O | \lambda)} \]

\[ \xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_k \sum_l \alpha_t(k) a_{kl} b_l(O_{t+1}) \beta_{t+1}(j)} \]

The denominator is \textbf{marginalized} over all states \( k \) at time \( t \) and all states \( l \) at time \( t+1 \)
Problem 4: Learning

\[ \xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_k \sum_l \alpha_t(k) a_{kl} b_l(O_{t+1}) \beta_{t+1}(j)} \]

- \( \alpha_t(i) \): it explains the first \( t \) observations and ends in state \( S_i \) at time \( t \) (green).
- \( a_{ij} \): it explains the probability of transitioning from \( S_i \) at time \( t \) to state \( S_j \) at time \( t+1 \)
- \( b_j(O_{t+1}) \): it generates the probability of observation from state \( S_j \) at time \( t+1 \) (red)
- \( \beta_{t+1}(j) \): it explains the probability of generating the rest of the observations \( (O_{t+2}, \ldots, O_1) \) from state \( S_j \) at time \( t+1 \) (blue)

Normalization is done by dividing for all such possible pairs that can be visited at times \( t \) and \( t+1 \).
Problem 4: Learning

- Now based on the derived expression for $\xi_t(i,j)$, we summarize the EM steps of the Baum-Welch algorithm:

\[
\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_k \sum_l \alpha_t(k) a_{kl} b_l(O_{t+1}) \beta_{t+1}(j)}
\]

\[
\gamma_t(i) \equiv \sum_{j=1}^{N} \xi_t(i,j)
\]

**E-step:**
1. Initialize the model $\lambda = (A, B, \Pi)$.
2. Compute $\xi_t(i,j)$ and $\gamma_t(i)$.

**M-step:**
3. Re-estimate the model $\lambda = (A, B, \Pi)$.
4. If $P(O | \lambda)$ increases, go to step 2 (or quit after a max number of iterations)