10-701

Machine Learning

Learning in HMMs
A Hidden Markov model

- A set of states \{s_1 \ldots s_n\}
  - In each time point we are in exactly one of these states denoted by \(q_t\)
- \(\Pi_i\), the probability that we *start* at state \(s_i\)
- A transition probability model, \(P(q_t = s_i \mid q_{t-1} = s_j)\)
- A set of possible outputs \(\Sigma\)
  - At time \(t\) we emit a symbol \(\sigma \in \Sigma\)
- An emission probability model, \(p(o_t = \sigma \mid s_i)\)
Inference in HMMs

• Computing $P(Q)$ and $P(q_t = s_i)$

• Computing $P(Q \mid O)$ and $P(q_t = s_i \mid O)$

• Computing $\text{argmax}_Q P(Q)$
Most probable path

• We are almost done …
• One final question remains
  How do we find the most probable path, that is Q* such that
  \[ P(Q^* \mid O) = \text{argmax}_Q P(Q \mid O) \]?

• This is an important path
  - The words in speech processing
  - The set of genes in the genome
  - etc.
Most probable path

\[ \arg \max_{q} P(Q \mid O) = \arg \max_{q} \frac{P(O \mid Q)P(Q)}{P(O)} \]

\[ = \arg \max_{q} P(O \mid Q)P(Q) = \arg \max_{q} P(O, Q) \]

We will use the following definition:

\[ \delta_t(i) = \max_{q_1 \ldots q_{t-1}} p(q_1 \ldots q_{t-1} \land q_t = s_i \land O_1 \ldots O_t) \]

In other words we are interested in the most likely path from 1 to t that:

1. Ends in \( S_i \)
2. Produces outputs \( O_1 \ldots O_t \)
Computing $\delta_t(i)$

$$
\delta_1(i) = p(q_1 = s_i \land O_1)
= p(q_1 = s_i)p(O_1 | q_1 = s_i)
= \pi_i b_i(O_1)
$$

$$
\delta_t(i) = \max_{q_1...q_{t-1}} p(q_1...q_{t-1} \land q_t = s_i \land O_1...O_t)
$$

Q: Given $\delta_t(i)$, how can we compute $\delta_{t+1}(i)$?

A: To get from $\delta_t(i)$ to $\delta_{t+1}(i)$ we need to

1. Add an emission for time $t+1$ ($O_{t+1}$)

2. Transition to state $s_i$

$$
\delta_{t+1}(i) = \max_{q_1...q_t} p(q_1...q_t \land q_{t+1} = s_i \land O_1...O_{t+1})
= \max_j \delta_t(j)p(q_{t+1} = s_i | q_t = s_j) p(O_{t+1} | q_{t+1} = s_i)
= \max_j \delta_t(j)a_{j,i}b_i(O_{t+1})
$$
The Viterbi algorithm

\[
\delta_{t+1}(i) = \max_{q_1 \cdots q_t} p(q_1 \cdots q_t \land q_{t+1} = s_i \land O_1 \cdots O_{t+1}) \\
= \max_j \delta_t(j)p(q_{t+1} = s_i \mid q_t = s_j)p(O_{t+1} \mid q_{t+1} = s_i) \\
= \max_j \delta_t(j)a_j,i b_i(O_{t+1})
\]

• Once again we use dynamic programming for solving \( \delta_t(i) \)
• Once we have \( \delta_t(i) \), we can solve for our \( P(Q^* \mid O) \)

By:

\[
P(Q^* \mid O) = \arg\max_Q P(Q \mid O) = \text{path defined by } \arg\max_j \delta_T(j),
\]
Inference in HMMs

- Computing $P(Q)$ and $P(q_t = s_i)$
- Computing $P(Q | O)$ and $P(q_t = s_i | O)$
- Computing $\text{argmax}_QP(Q)$
Learning HMMs

- Until now we assumed that the emission and transition probabilities are known
- This is usually not the case
  - How is “AI” pronounced by different individuals?
  - What is the probability of hearing “class” after “AI”?

While we will discuss learning the transition and emission models, we will not discuss selecting the states.

This is usually a function of domain knowledge.
Example

• Assume the model below
• We also observe the following sequence:
  1,2,2,5,6,5,1,2,3,3,5,3,3,2 ..... 
• How can we determine the initial, transition and emission probabilities?
Initial probabilities

Q: assume we can observe the following sets of states:

AAABBA
AABBBB
BAABBAB

how can we learn the initial probabilities?

A: Maximum likelihood estimation

Find the initial probabilities $\pi$ such that

$$
\pi^* = \arg \max_\pi \prod_k \pi(q_1) \prod_{t=2}^T p(q_t | q_{t-1})
$$

$$
\pi^* = \arg \max_\pi \prod_k \pi(q_1)
$$

$$
\pi_A = \#A / (\#A + \#B)
$$

k is the number of sequences available for training
Q: assume we can observe the set of states:
AAABBBAAAAABBBBBAAABBBB

how can we learn the transition probabilities?

A: Maximum likelihood estimation

Find a transition matrix $a$ such that

$$a^* = \arg \max_a \prod_k \pi(q_1) \prod_{t=2}^T p(q_t | q_{t-1})$$

$$a^* = \arg \max_a \prod_{t=2}^T p(q_t | q_{t-1})$$

$$a_{A,B} = \#AB / (\#AB + \#AA)$$

Remember that we defined $a_{i,j} = p(q_t = s_j | q_{t-1} = s_i)$
Q: assume we can observe the set of states:

A A A B B A A A A B B B B B A A

and the set of dice values

1 2 3 5 6 3 2 1 1 3 4 5 6 5 2 3

how can we learn the emission probabilities?

A: Maximum likelihood estimation

\[ b_A(5) = \frac{\#A5}{\#A1 + \#A2 + \ldots + \#A6} \]
Learning HMMs

• In most case we do not know what states generated each of the outputs (fully unsupervised)

• … but had we known, it would be easy to determine an emission and transition model

• On the other hand, if we had such a model we could determine the set of states using the inference methods we discussed
Expectation Maximization (EM)

• Appropriate for problems with ‘missing values’ for the variables.
• For example, in HMMs we usually do not observe the states
Expectation Maximization (EM): Quick reminder

- Two steps
- E step: Fill in the expected values for the missing variables
- M step: Regular maximum likelihood estimation (MLE) using the values computed in the E step and the values of the other variables
- Guaranteed to converge (though only to a local minima).
Forward-Backward

• We already defined a *forward* looking variable

\[ \alpha_t(i) = P(O_1 \cdots O_t \land q_t = s_i) \]

• We also need to define a *backward* looking variable

\[ \beta_t(i) = P(O_{t+1}, \cdots, O_n \mid s_t = i) \]
Forward-Backward

• We already defined a *forward* looking variable

\[ \alpha_t(i) = P(O_1 \ldots O_t \land q_t = s_i) \]

• We also need to define a *backward* looking variable

\[ \beta_t(i) = P(O_{t+1}, \ldots, O_n \mid s_t = i) = \sum_j a_{i,j} b_j (O_{t+1}) \beta_{t+1}(j) \]
Forward-Backward

• We already defined a *forward* looking variable

\[ \alpha_t(i) = P(O_1 \ldots O_t \land q_t = s_i) \]

• We also need to define a *backward* looking variable

\[ \beta_t(i) = P(O_{t+1}, \ldots, O_n \mid s_t = i) \]

• Using these two definitions we can show

\[
P(q_t = s_i \mid O_1, \ldots, O_n) = \frac{\alpha_t(i) \beta_t(i)}{\sum_j \alpha_t(j) \beta_t(j)} = S_t(i)
\]

\[ P(A \mid B) = P(A, B) / P(B) \]
State and transition probabilities

• Probability of a state

\[
P(q_t = s_i \mid O_1, \cdots, O_n) = \frac{\alpha_t(i) \beta_t(i)}{\sum_j \alpha_t(j) \beta_t(j)} = S_t(i)
\]

• We can also derive a transition probability

\[
P(q_t = s_i, q_{t+1} = s_j \mid o_1, \cdots, o_n) = S_t(i, j)
\]

\[
P(q_t = s_i, q_{t+1} = s_j \mid o_1, \cdots, o_n) = \frac{\alpha_t(i) P(q_{t+1} = s_j \mid q_t = s_i) P(o_{t+1} \mid q_{t+1} = s_j) \beta_{t+1}(j)}{\sum_j \alpha_t(j) \beta_t(j)} = S_t(i, j)
\]
E step

- Compute $S_t(i)$ and $S_t(i,j)$ for all $t, i, \text{ and } j \ (1 \leq t \leq n, 1 \leq i \leq k, 2 \leq j \leq k)$

$$P(q_t = s_i \mid O_1, \ldots, O_n) = S_t(i)$$

$$P(q_t = s_i, q_{t+1} = s_j \mid o_1, \ldots, o_n) = S_t(i, j)$$
M step (1)

Compute transition probabilities:

\[ a_{i, j} = \frac{\hat{n}(i, j)}{\sum_k \hat{n}(i, k)} \]

where

\[ \hat{n}(i, j) = \sum_t S_t(i, j) \]
M step (2)

Compute emission probabilities (here we assume a multinomial distribution):

define:

\[ B_k(j) = \sum_{t|o_t=j} S_t(k) \]

then

\[ b_k(j) = \frac{B_k(j)}{\sum_i B_k(i)} \]
Complete EM algorithm for learning the parameters of HMMs (Baum-Welch)

- Inputs: 1. Observations $O_1$ ... $O_n$
- 2. Number of states, model

1. Guess initial transition and emission parameters
2. Compute E step: $S_t(i)$ and $S_t(i,j)$
3. Compute M step
4. Convergence? No
5. Output complete model

We did not discuss initial probability estimation. These can be deduced from multiple sets of observation (for example, several recorded customers for speech processing)
Building HMMs—Topology

Matching states

Insertion states

Deletion states

No of matching states = average sequence length in the family
PFAM Database - of Protein families
(http://pfam.wustl.edu)
A HMM model for a DNA motif alignments, The transitions are shown with arrows whose thickness indicate their probability. In each state, the histogram shows the probabilities of the four bases.