HMM Lecture Notes
Dannie Durand and Rose Hoberman

Hidden Markov Models

In today’s lecture, we discussed the recognition problem with HMMs and one method used to solve it: the Viterbi algorithm.

Review

HMMs are defined formally as follows:

1. N states $S_1...S_N$
2. M symbols in alphabet
3. Transition probability matrix $a_{ij}$
4. Emission probabilities $e_i(a)$ probability state $i$ emits character $a$.
5. Initial distribution vector $\pi = (\pi_i)$

We refer to the emission probabilities, the transition probabilities and the initial distribution, collectively as the parameters, designated $\lambda = (a_{ij}, e_i(a), \pi)$. Following the notation used in Durbin, we will refer to the sequence of observed symbols as $O = O_1, O_2, O_3,...$ and the sequence of states visited as $Q = q_1, q_2, q_3,...$ (the “state path”). When considering more than one sequence or state path, we will use superscripts to distinguish them: $O^i = O^i_1, O^i_2, O^i_3,...$ and $Q^j = q^j_1, q^j_2, q^j_3,...$

The parameters of an HMM can be learned from labeled data:

$$a_{ij} = \frac{A_{ij}}{\sum_{j'} A_{ij'}} \quad e_i(x) = \frac{E_i(x)}{\sum_{x'} E_i(x')}$$

where $A_{ij}$ is the number of transitions from state $S_i$ to $S_j$ in the training data and $E_i(x)$ is the number of instances of the symbol $x$ that are labeled with state $S_i$. Pseudocounts can be used to capture symbols and states not captured in the training data.

In an HMM, each state emits a symbol from a fixed alphabet each time a state is visited. Emission probabilities are state-dependent, but not time-dependent. A symbol may be emitted by more than one state. Similarly, a state can emit more than one symbol.

Note that an HMM is a generative model. It gives the probability of generating a particular sequence (hence, the emission probabilities.) In an HMM, there may be more than one, and often
very many, state paths associated with $O$. Therefore, the “true” sequence of states that generated the observed sequence is unknown, or hidden, hence the term, “Hidden” Markov model. The state path is hidden because it is not possible to tell the state merely by the output symbol. This hidden sequence of states corresponds to what we want to know, namely the classification of each symbol.

Given a sequence sequence, $O$, there are several questions we may wish to ask:

1. What is the true path? Otherwise stated, we wish to assign labels to an unlabeled sequence.
   
   **Example:** Identify the cytosolic, transmembrane, and extracellular regions in the sequence.
   
   In this case, we wish to assign the labels E, M, or C to the unlabeled data.

2. What is the probability that a given sequence $O$, was generated by the HMM?
   
   **Example:** Is the sequence a transmembrane protein?

**The Viterbi algorithm:**

In the transmembrane example, the amino acid sequence is known, but the subcellular location of each residue is hidden. In the HMM model, the subcellular location of each residue is represented by the (hidden) state that emitted the symbol associated with that residue. We will infer the subcellular locations of the residues by inferring the sequence of states visited by the HMM. The boundaries between the transmembrane, extracellular and cytosolic regions are defined by the transitions between $C$, $M$, and $E$ states along this state path. This process is called “decoding” because we decode the sequence of symbols to determine the hidden sequence of states. HMMs were developed in the field of speech recognition, where recorded speech is “decoded” into words or phonemes to determine the meaning of the utterance.

There are two common approaches to decoding: Viterbi decoding and Posterior decoding. Viterbi decoding is based on the assumption that most likely path, $Q^* = \arg \max_Q P(Q|O)$, is a good estimation of the sequence of states that generated a given observed sequence $O$. In posterior decoding, which we will discuss next week, the sequence of most probable states is used to estimate true path. Note that $q_t^*$, the $t$th state on the most probable path, is not necessarily the state most likely to be visited at time $t$, when all possible paths are considered.

In today’s lecture, we discussed the Viterbi decoding. We could use brute force by calculating $P(Q|O)$ for all paths. However, this becomes intractable as soon as number of states gets larger, since the number of state sequences grows exponentially ($N^T$). Instead, we calculate $\arg \max_Q P(Q,O)$ using a dynamic programming algorithm called the Viterbi algorithm. Note, that this will still give us the most probable path because the path that maximizes $P(Q,O)$ also maximizes $P(Q|O)$:

$$\arg \max_Q P(Q|O) = \arg \max_Q \frac{P(Q,O)}{P(O)} = \arg \max_Q P(Q,O).$$
Let $\delta(t, i)$ be the probability of observing the first $t$ residues and ending up in state $S_i$ via the most probable path. We calculate $\delta(t, i)$ as follows:

**Initialization:**

$$
\delta(1, i) = \pi_i e_i(O_1)
$$

**Recursion:**

$$
\delta(t, i) = \max_{1 \leq j \leq N} \delta(t - 1, j) \cdot a_{ji} \cdot e_i(O_t)
$$

**Final:** The probability of the most probable path

$$
P(Q^*) = \max_{1 \leq i \leq N} \delta(T, i)
$$

The final state, $q_T$ is the state that maximizes $\delta(T, i)$. The state path can be reconstructed by tracing back through the dynamic programming matrix, similar to the traceback in pairwise sequence alignment.

**Running time:** $O(TN^2)$

There are $TN$ entries in the dynamic programming matrix. Each entry requires calculating $N$ terms.

**HMM’s and probabilities**

An HMM emits each sequence $O^i \in \Sigma^*$ with probability $P(O^i) \geq 0$. Every time an HMM is executed it will emit a sequence. Therefore,

$$
\sum_i P(O^i) = 1
$$

Since a sequence, $O$ can, potentially, be emitted from more than one state path, to obtain the total probability of a sequence we must sum over the all possible paths:

$$
P(O) = \sum_j P(O|Q^j) \cdot P(Q^j).
$$

Fig. 1 shows $P(O, Q)$ for the set of all possible state paths, $Q$. The most likely path is the maximum point on this curve and is marked with a star. The total probability of sequence $O$, $P(O)$, is equal to the area under the curve.
Figure 1: The probability of sequence $O$ and state path $Q$. The area under this curve is $P(O)$. The maximum point on the curve is the most likely path, $Q^*$. From this, we obtain

$$\sum_i P(O^i) = 1$$

$$\sum_i \sum_j P(O^i|Q^j) \cdot P(Q^j) = 1$$

$$\sum_i \sum_j P(O^i, Q^j) = 1$$
An example

In class, we used the three-state HMM shown in Fig. 2 as an example. The parameters for this model are

\[
\begin{array}{l}
\pi_i & 0.5 & 0 & 0.5 \\
\epsilon_i(H) & 0.2 & 0.9 & 0.3 \\
\epsilon_i(L) & 0.8 & 0.1 & 0.7 \\
\end{array}
\]

As an exercise, try applying the Viterbi algorithm to determine the most likely path through the three-state HMM above for the sequence \textit{HHHH}. A worksheet for this exercise is linked to the class syllabus page. The solution is also available. I recommend that you try to work through the Viterbi algorithm before looking at the solution.