Reading for Lecture 9

Release v10

Christopher Lee

October 17, 2011

Contents

1 Posterior Probability Calculation with the Forward-Backward Algorithm **i**
- 1.1 Forward Probability Calculation . . . i
- 1.2 Backward Probability Calculation . . . ii
- 1.3 Posterior Probability Calculation . . . ii

2 Heterogeneous HMMs **iii**
- 2.1 Heterogeneous HMM Characteristics . . . iii
- 2.2 What's Different? . . . iii
- 2.3 What's the Same? . . . iii

3 Length Variation in an HMM **iv**
- 3.1 Ways of Modeling Length Variation . . . iv
- 3.2 Length Distribution of Basic Model Structures . . . iv

Read Jones & Pevzner sections 11.4 - 11.6, and the following material.

1 Posterior Probability Calculation with the Forward-Backward Algorithm

1.1 Forward Probability Calculation

Say we wish to compute the posterior probability

$$p(\theta_t|\vec{O}^n) = \frac{p(\theta_t, \vec{O}^n)}{p(\vec{O}^n)}$$

The obvious difficulty here is that this depends on the entire observation sequence, which under the general chain rule would require summing the expression above over all the hidden variables, an $O(m^n)$ computation. Can the Markov property make this easier? The Viterbi algorithm suggests that it can, and indicates a general strategy: find a recursion that takes advantage of the Markov property, that can give us what we want. Following the idea of the

Viterbi algorithm, let's consider just the observations $O_1, O_2, ...O_t$, which we write as $\vec{O}^t$. We now seek a recursion to compute $p(\theta_t, \vec{O}^t)$ in terms of its predecessor $p(\theta_{t-1}, \vec{O}^{t-1})$:

$$p(\theta_t, \vec{O}^t) = \sum_{\theta_{t-1}} p(\theta_{t-1}, \vec{O}^{t-1}) p(\theta_t|\theta_{t-1}) p(O_t|\theta_t)$$

This shows the power of the Markov property: each step of this calculation (e.g. $t-1 \to t$) only takes $O(m^2)$ time. So we could use this calculation to obtain $p(\vec{O}^n)$ in just $O(nm^2)$ time. This is called a "forward probability" calculation since it must be done in order $\theta_1, \theta_2, ...$ Note that the first step is just:

$$p(\theta_1, \vec{O}^1) = p(\theta_1) p(O_1|\theta_1)$$

and the last step is:

$$p(\vec{O}^n) = \sum_{\theta_n} p(\theta_n, \vec{O}^n)$$

1.2 Backward Probability Calculation

Since

$$p(\theta_t, \vec{O}^n) = p(\theta_t, \vec{O}^t) p(\vec{O}^n_{t+1}|\theta_t)$$

all we need to complete our calculation is $p(\vec{O}^n_{t+1}|\theta_t)$. Again, we seek a recursion to compute it in terms of its "follower" $p(\vec{O}^n_{t+2}|\theta_{t+1})$:

$$p(\vec{O}^n_{t+1}|\theta_t) = \sum_{\theta_{t+1}} p(\vec{O}^n_{t+2}|\theta_{t+1}) p(O_{t+1}|\theta_{t+1}) p(\theta_{t+1}|\theta_t)$$

This is called a "backward probability" calculation since it must be done in reverse order $\theta_n, \theta_{n-1}, ...$ Note that the first step is just

$$p(\vec{O}^n_{n+1}|\theta_n) = 1$$

since $\vec{O}^n_{n+1}$ is just an empty sequence, i.e. there is no observable variable with an index higher than n.

1.3 Posterior Probability Calculation

Finally, we use the results of both calculations to compute the posterior probability of θ_t:

$$p(\theta_t|\vec{O}^n) = \frac{p(\theta_t, \vec{O}^n)}{p(\vec{O}^n)} = \frac{p(\theta_t, \vec{O}^t) p(\vec{O}^n_{t+1}|\theta_t)}{p(\vec{O}^n)}$$

Note that this calculation takes $O(nm^2)$ to compute a single posterior $p(\theta_t|\vec{O}^n)$. However, we can compute *all* the $p(\theta_t|\vec{O}^n)$ (i.e. for all values of t), by computing all the forward probabilities (an $O(nm^2)$ computation), and all the backward probabilities (also $O(nm^2)$), which together give us all the $p(\theta_t|\vec{O}^n)$.

2 Heterogeneous HMMs

Up to this point we have restricted our attention to *homogeneous HMMs* where the same states and transition matrix are used to model every hidden variable θ_t in the Markov chain. By contrast, in a *heterogeneous HMM* the transition probabilities change at different times, or, equivalently, the set of accessible states and transitions changes permanently as a function of time.

Example: a Protein Sequence Profile

Say we have a large set of aligned sequences from a single family of related proteins. At some positions in the alignment we may see complete conservation (i.e. only a single amino acid is observed at this position, in all members of the family), whereas other positions may only permit a few alternative amino acids, whereas other positions may be very different in different members of the family. Whereas a traditional sequence alignment scoring matrix always assigns the same set of "mismatch scores" to a given amino acid no matter what position in a protein it occurs at, such a family alignment shows that it should perhaps be scored very differently at different positions. For example, at positions where that amino acid is totally conserved, any mismatch should be scored as extremely unlikely, whereas at positions where that amino acid is frequently observed to be replaced by another amino acid, a mismatch might actually be more likely than a match.

To model this fine-grained scoring, we can construct an HMM where each variable (representing one position in the protein family sequence) has a *different* set of states:

- a "profile state" whose emission probabilities simply match the observed frequencies of different amino acids seen at this position in the family alignment;
- deletion and insertion states that enable the HMM to skip or "linger" at this position (i.e. either to emit no letter corresponding to this position, or more than one; for further details see the section on variable length distributions below).

See Jones & Pevzner for further details on profile HMMs.

2.1 Heterogeneous HMM Characteristics

- Each variable θ_t in the Markov chain typically is represented by a group of nodes (states) that represent all of its possible values.
- Typically each such group has transitions to states representing the next variable (i.e. θ_{t+1}), and possibly to itself (i.e. a "self-edge").
- In general there are no edges to states representing "previous" variables in the Markov chain. Thus the Markov path is forced to proceed from START to END with no backtracking. In other words, once we leave a group of states, we never return.
- Note that this is fundamentally different from what we have typically assumed about homogeneous HMMs, namely that every state is accessible from every other state.

2.2 What's Different?

- The whole idea of a "stationary distribution" goes out the window when a different set of states and transitions is supplied for each new step in the Markov chain.

2.3 What's the Same?

Otherwise, there aren't very many changes. Most Markov chain properties and algorithms still apply, as long as they don't require convergence to a stationary distribution. In particular, the Viterbi and forward-backward algorithms still

work, without modification.

3 Length Variation in an HMM

So far we have discussed problems where the length of the HMM (i.e. the number of variables) was itself not a variable. For example, in modeling CpG islands, the length of individual CpG islands was allowed to vary, but the length of the whole HMM was fixed to match the length of the genomic sequence being modeled. However, there are many problems where the length of the HMM itself must be allowed to vary.

Example: a Protein Sequence Profile

Sequences of related proteins often vary considerably in length, either by leaving out certain segments altogether, or by allowing insertion or deletion of individual letters at various points in the sequence. If we construct an HMM "profile" for a specific protein family, it must be able to emit / model a wide variety of "observed" sequence lengths.

3.1 Ways of Modeling Length Variation

- *insertion states*: these are states that emit one or more "extra" letters between two consecutive positions in the consensus profile. Moreover, insertion states typically have self-edges that permit the path to remain in that specific insertion state for any number of steps (inserting that number of letters in the emitted output).
- *deletion states*: these are states that permit the HMM to skip a consensus position in the profile. Rather than adding edges from every consensus position to every other consensus position (an $O(N^2)$ number of edges), we instead create one deletion state for each consensus position. The deletion state emits no observation, so an HMM path that passes through the deletion state instead of the consensus state in effect skips that consensus position. Furthermore, deletion states for consecutive positions are connected to each other as well as to the next consensus position, so paths can skip any number of consensus positions in a row.

Another form of length variation consists of multiple entry or exit points in the HMM:

- If the START node has a outgoing edges to states for only the first variable in the HMM (i.e. θ_1), then it does not contribute to length variation. On the other hand, if it has outgoing edges to many different variables, it can both create a wide variety of lengths, and generate any desired distribution of length probabilities. That is, the probability of each entry point to the HMM is controlled separately by the edge weight associated with the edge(s) from START to that entry point.
- Similarly, if multiple variables in the model have edges to the END state, that again provides fine control over the length of the path through the HMM (recall that once the path goes to the END state, it terminates). Each "exit edge" can be assigned a different transition probability, to give whatever probability distribution is desired for the different "exit points" and associated lengths.

3.2 Length Distributions of Basic Model Structures

- Consider a single node with a self-edge with probability p. The path through this node can be one to ∞ letters long, with a length distribution

$$p(L) = p^{L-1}(1-p)$$

  This is an exponential decay: the most likely length is $L = 1$, and the probability goes down by a factor of p for each additional letter.

- To create a continuum of distribution curves from exponential decay to bell-shaped (normal), we can simply chain together two or more identical nodes with self-edges. If the number of nodes in the chain is C, the length distribution is

$$p(L) = \binom{L-1}{C-1} p^{L-C}(1-p)^{C}$$

  For $C = 1$ the curve is pure exponential decay; as C increases it becomes gradually more bell-shaped.