Hidden Markov Models

Philipp Ross

2/2/2017

Introduction to HMMs

In simple Markov models we have a set of observed states, say $X_n, n = 1, 2, ...$ with transition probabilities $p_{i \to j}$ of changing from state i to state j. But what if the states are not directly observed? We might run into a situation where we directly observe the outputs of a process that are probabilistically determined depending on which state the process is in, but we don't directly observe the states. **Hidden Markov Models** become applicable in this scenario.

Below is a graphical representation:

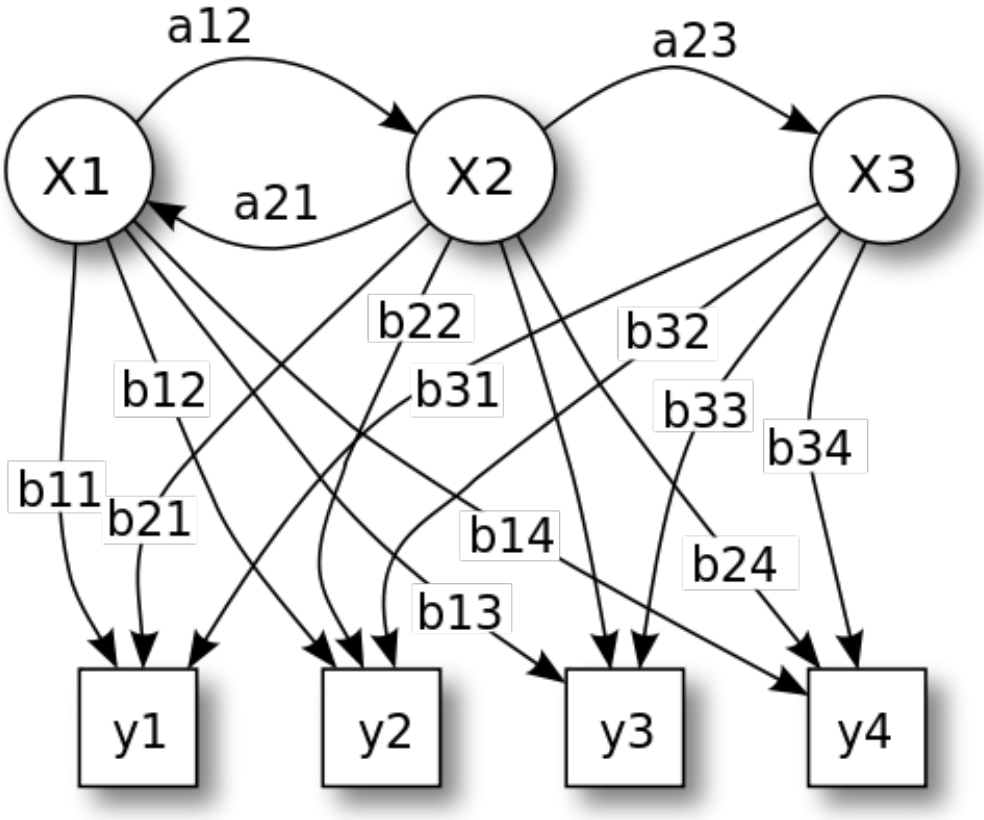

Figure 1:

Figure 1. Probabilistic parameters of a hidden Markov model (example)
X — states
y — possible observations

a — state transition probabilities
b — output probabilities

An example of a hidden Markov process is a generalization of the Urn problem without replacement. Imagine a room of urns X_1, X_2, X_3 each with a mixture of different balls labeled y_1, y_2, y_3. There's someone inside the room choosing balls from the urn and showing you only the labelled balls, but not the urn from which they came. Additionally, assume there is some probabilistic process of choosing which urn to sample from, that the choice of the next urn only depends directly on the urn that was chosen before it, and that you know the relative proportions of different balls within each urn. This process meets the criteria for a hidden markov process.

A biologically relevant example of where HMMs are used is in genome sequence annotation. For example, in gene prediction, you have some prior information about what a gene looks like, hidden states that represent whether you have a gene or not, a probability of moving between non-protein coding and protein-coding sequence, and a set of observations that are the sequences themselves.

Lecture Notes

HMMs model an underlying sequence of observations. These observations are emitted by hidden states.

We are interested in the following random variables:

$Q = (Q_1, ..., Q_N)$, $O = (O_1, ..., O_K)$, $A = (a_{ij}, i = 1, ..., N, j = 1, ..., N)$, and $\pi = (pi_i, ..., \pi_N)$ where Q are the hidden states, O are the potential observations, A are state transition probabilities, and π are the prior probabilities or weights.

While these are necessary for defining the hidden layer of the hidden markov model, we still need to define emission probabilities. That is, the probability that we get a certain output based on being in a certain state at time t. We define these as:

$$B = b_{ij}, i = 1, ..., N, j = 1, ..., K$$

Say $\lambda = (A, B, \pi)$

In order to impliment an HMM we must first solve the following problems:

1. We need to compuate the liklehood, $L(\lambda) = P(O|\lambda)$
2. We need to compute the maximum likelihood estimate for λ, $\widehat{\lambda} = argmax P(O|\lambda)$
3. Finally, we need to compute the maximum likelihood estimate of Q, $\widehat{\underset{\lambda}{\widehat{Q}}} = \underset{Q}{argmax} P(Q|O, \lambda)$

Four well known algorithms can used to solve these problems:

1. Forward algorithm
2. Backward algorithm
3. Virterbi algorithm (example of dynamic programming)
4. Baum-Welch algorithm (example of an EM algorithm)

Calculating the likelihood

We want the marginal distribution of O to calculate:

```
$$
\begin{align*}
P(O|\lambda) &= \sum_{Q} P(O,Q|\lambda) \\
             &= \sum_{Q} P(Q_{1})P(O_{1}|Q_{1})P(Q_{2}|Q_{2})P(O_{2}|Q_{2})...P(O_{T}|Q_{T}) \\
             &= \sum_{Q} q \pi_{q_{1}} b_{q_{1}}(O_{1}) a_{q_{1}q_{2}} b_{q_{2}}(O_{2}) a_{q_{2}q_{3}} b
\end{align*}
$$
```

For this naive summation we would have N^T terms. However, if we distribute the sums similar to Feselstein's pruning algorithm for calculating the likelihoods of phylogenetic trees, we can hopefully reduce the number of terms.

Forward algorithm

1. Initialize (N steps)

$\alpha_1(i) = \pi_i b_i(O_1)$

Then you get $\sum_{q_1} \alpha(q_1) a_{q_1 q_2}$ because there are no other q_1s

2. Compute $\alpha_t(i)$ recursively for $t = 1...T$ (TN^2 steps)

$\alpha_t(i) = \sum_j \alpha_{t-1}(j) a_{ji} b_i(O_t)$

$\sum_{q_1} \alpha_1(q_1) a_{q_1 q_2} b_{q_2}(O_2)$

$\sum_{q_3 \ldots q_T} \sum_{q_2} \alpha_2(q_2) a_{q_2 q_3} b_{q_3}(O_3) a_{q_3 q_4} ... b_{q_T}(O_T)$

$\alpha_t(i) = P(O_1 ... O_T, Q_t = i)$

3. Calculate $\sum_i \alpha_T(i) = P(O|\lambda)$ (N steps)

Now the computation is approximately on the order of TN^2

Backward algorithm

Perform the forward algorithm in reverse

1. $B_T(j) = 1, j = 1, ..., N$
2. $B_t(j) = \sum_i a_{ji} b_i(O_t) B_{t+1}(i)$

$\alpha_t(i) = P(O_1 ... O_T, Q_t|\lambda)$ $B_t(i) = P(O_{t+1}, ... O_T|Q_t = i|\lambda)$

If we are interested if a particular region of the genome is "open", $P(Q_t = i|O_i, \lambda) = \frac{P(O_t, Q_t = i|\lambda)}{P(O|\lambda)}$ where the numerator is the product of $\alpha_t(i)$ and $B_t(i)$ and the denominator is $\sum_i \alpha_t(i) B_t(i)$

So how do we calculate $\underset{Q}{argmax} P(Q|O, \lambda)$?

$\underset{Q}{argmax} P(Q|O, \lambda) = argmax \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) a_{q_2 q_3} b_{q_3}(O_3) ... b_{q_T}(O_T)$

Break this up into subproblems... successive set of maximization problems. Now we have T different maximizations.

$\delta_1(q_2) = max q_1 \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2}$ $\delta_2(q_3) = max q_2 \delta_{q_1}(q_2) b_{q_2}(O_2) q_{q_2 q_3}$ . . . $delta_T(q_T)$

Again, we get an algorithm with complexity of TN^2

Lattice model is a different way of visualizing a markov chain.

Baum-Welch algorithm

Example of an EM algorithm.

$\theta^{t+1} = argmax E_{z|x}[logP(X, Z|\theta)]$

Here we have

$\lambda^{t+1} = argmax E_{Q|O}[\lambda^t] logP(O, Q|\lambda)$

$\sum_Q logP(O, Q|\lambda) P(Q|O, \lambda^t)$

$\sum log(\pi_i) I_Q = i + \sum log(a_{ij}) \cdot \#t_{ij}(Q) + \sum log(b_{ik}) \cdot \#O_{ij}(Q) P(Q|O, \lambda^t)$

Where $\#t_{ij}(Q)$ = number of transitions from i to j in path Q.

$\sum_{i,j} E[\#t_{ij}(Q)|\lambda^t, O] log(a_{ij}) + \sum_{i,j} E[\#O_{ij}(Q)|\lambda^t, O] log(b_{ij}(j)) + \sum P(Q = i|lambda^t, O) log(\pi_i)$

```
$$
\begin{align*}
a_{ij}^{t+1} &= \frac{E[\# t_{ij}(Q)|\lambda^{t},O]}{E[\# of times in state i|\lambda^{t},O]} \\
b_{ij}^{t+1} &= \frac{E[\# O_{ij}(Q)|\lambda^{t},O]}{E[\# times in state i|\lambda^{t},O]} \\
pi_{i}^{t+1} &= P(Q_{i}=1|\lambda^{t},O) \\
end{align*}
$$
```

Helpful links

1. Rabiner Tutorial
2. Wikipedia article on HMMs
3. Nature Biotechnology Primer on HMMs