Lecture 11: Efficient Methods for Training HMMs

Last time we saw an instance of the EM algorithm, where we used an initial probability distribution for hidden data to generate a new corpus of weighted data that was "fully tagged" and thus we could re-estimate a new probability distribution that is better (or the same) as the initial estimate. The problem with the method is, of course, that the technique can require an exponential expansion of the corpus in general. When trying to train HMM models, and even fairly simple HMMs can produce a large number of paths in practice. For instance, the HMM in Figure 1 is the same as that last time but we removed the start and end states and fully connected them. Rather than the 3 possible sequences from last time, there are know eight possible sequences that generate R W B B. In general, with N states we will have N^T paths where T is the length of the observed sequence. So we need to develop ways to calculate the necessary values efficiently. This algorithm is called the **Baum-Welch reestimation method** or the **forward-backward algorithm**. Rather than enumerating the paths, this method "counts" by looking at the probabilities of reaching various states as well as probabilities of completing the sequence starting from a given state. These are called the forward and backward probabilities, respectively, and we will develop them first before looking at their use in reestimation.

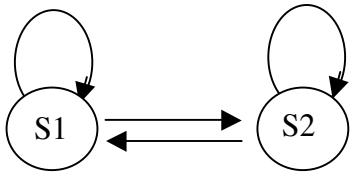

Figure 1: An example HMM for Training

The key to doing this efficiently is to be able to compute the probability of the HMM being in a particular state at a particular time for the output. Let the observed sequence be $O_{1,T}$, and let Q_i be random variables that produce the state that the HMM is in at time i. Given this, we are interested in

$$P(Q_t=S_i, O_1, \ldots, O_T)$$

We will access this probability by dividing it into two parts, using the chain rule that allows us to rewrite it as

$$= P(Q_t=S_i, O_1, .., O_t)*P(O_{t+1}, \ldots, O_T \mid Q_t=S_i, O_1, .., O_t)$$

Which, because the Markov assumption says that the future output depends only on the current state,

$$= P(Q_t=S_i, O_1, .., O_t)*P(O_{t+1}, \ldots, O_T \mid Q_t=S_i)$$

These two terms are called the **forward** and **backward** probabilities and we will spend most of today learning how to estimate them.

1. Computing Forward Probabilities

Then the **forward probability** for state i at time t, $\alpha_t(i)$, is the probability that an HMM will output a sequence $O_{1,t}$ and end in state s_i. For instance, the forward probability of being in state s_1 after two steps on the sequence *R W B B* is the joint probability

Transition from\to	S1	S2
S1	.6	.4
S2	.3	.7

(a) Initial Transition Probability Matrix $A_{i,j}$

Output Prob	R	W	B
S1	.3	.4	.3
S2	.4	.3	.3

(b) Output Probabilities

S1	.8
S2	.2

(c) Initial State Probability

Figure 2: The Initial Distributions for the HMM

$\alpha_2(1) = P(Q_2=s_1, O_1 = R, O_2 = W)$

We develop an algorithm to compute the forward probabilities that is very similar to the Viterbi algorithm.

Let's first calculate the probability of being in each state at time 1 (with output R) given the initial probability distribution shown in Figure 2. This is simply the probability of the state being the initial state times the probability of having output R.

$\alpha_1(1) = P(Q_1=S1) * P(O_1 = R| Q_1=S1) = .8*.3 = .24$
$\alpha_1(2) = P(Q_1=S2) * P(O_1=R| Q_1=S2) = .2 * .4 = .08$

To compute the forward probabilities at time 2, we just look at the increments from time 1 using the general formula for time t

$\alpha_t(i) = \sum_j \alpha_{t-1}(j) * P(Q_t=S_i | Q_{t-1}=S_j) * P(O_t | Q_t = S_i)$

In other words, we simply sum over all extensions from every possible state at time t-1 to state S_i. Applying this to compute the forward probability of S2 at time t we get

$\alpha_2(2) = (\alpha_1(1) * P(S2|S1) + \alpha_1(2) * P(S2|S2)) * P(W|S2)$
$= (.24 * .4 + .08 * .7) * .3 = .046$

Once we have all the forward probabilities for time 2, we can then compute the forward probabilities for time 3, and so on. The results of this calculation for the given HMM and output R W B B is shown in Figure 3.

Forward Probs	R	W	B	B
S1	.24	.067	.016	.0045
S2	.08	.046	.017	.0056

Figure 3: The Values for the Forward Probabilities

Note that since the input is given we often want to know what the probability of being in a state is at time t given the input. For instance, what is the probability of being in state S1 at time 3 given the input R W B? This conditional probability is

$P(Q_3 = S1 | R W B) = P(Q_3 = S1, R W B) / P(R W B)$

The numerator is just the forward probability. But how do we compute the denominator, which is the probability of seeing R W B given this HMM. Well this simply is the sum over all the

forward probabilities at time 3 (since we have to end up in one of the states!). So the probability of being in state S1 at time 3 (after seeing R W B) is

$$P(Q_3 = S1 \mid O_{1,3}) = \alpha_3(1) / \sum_i \alpha_3(i)$$

$$= .016 / (.016 + .017) = .485$$

2. Backward Probability

The other probability we need is called the **backward probability,** which is the probability of starting in state S_i at time t and generating the rest of the observation sequence $o_{t+1},..., o_T$.

$$\beta_t(i) = Prob(o_{t+1},..., o_T \mid Q_t = S_i)$$

The backward probabilities can be computed efficiently using an algorithm that is a simple "backwards" variant of the forward algorithm. Rather than starting at time 1, the algorithm starts at time T and then works backwards through the network from observation o_T down to o_{t+1}. The initial probabilities of being in state S_i at time T and generating nothing else is simply 1 since there is no more output in this context.

i.e., $\beta_T(i) = P(\varnothing \mid S_i) = 1$

The recurrence formula used in the iterative step is

$$\beta_t(i) = \sum_j A_{i,j} * Prob(o_{t+1} \mid S_j) * \beta_{t+1}(j)$$

Figure 4 shows the backwards probabilities computed for our example. We show the t=0 case as well, which involves a slightly different probably involving the initial state distribution rather than a transition probability.

State	RWBB (t=0)	WBB (t =1)	BB (t = 2)	B (t =3)	∅ (t =4)
S1	.0078	.0324	.09	.3	1
S2	.0024	.0297	.09	.3	1

Figure 4: The Backward Probabilities for the Example

3. Using Forward and Backwards Probabilities

With both the forward and backward probabilities defined, we can now define the probability of observing $o_1 ... o_T$ and being in state S_i at time t as follows:

$$Prob(o_1 ... o_T, q_t = S_i) = \alpha_t(i) * \beta_t(i)$$

i.e., its the probability of observing $o_1 ... o_t$ ending in state S_i, times the probability of observing $o_{t+1} ... o_T$ given that we start in S_i. With this, we can now compute the conditional probability of being in state S_i at time t given the observation sequence

$$Prob(q_t = S_i \mid o_1 ... o_T) = Prob(o_1 ... o_T, q_t = S_i) / Prob(o_1 ... o_T)$$

The numerator we just saw equals $\alpha_t(i) * \beta_t(i)$. The denominator can be computed in many different ways, all producing the same result. For instance, we could sum the forward probabilities of all states in the final position, i.e., $\sum_i \alpha_T(i)$, or we could sum over all states which the backward probability at time 0, $\sum_i \beta_0(i)$. Or we could sum $\alpha_t(i) * \beta_t(i)$ over all states for any

Standard Notations Used in the Literature

Much of the literature on HMMs and discussion of the forward probability use a specific notation. We state the forward algorithm in these terms here so oyu can become familiar with it. Before doing that, we present the standard notations for defining HMMs.

An **HMM** λ consists of

1. a set of N states $S = \{S_1, ..., S_N\}$;
2. a set of R output symbols $V = \{v_1, ..., v_R\}$;
3. a k by k matrix $A_{i,j}$ which are the transition (i.e., $A_{i,j}$ = Pr(S_j follows state S_i);
4. an output probability distribution B specifying the probability of each output symbol and each state (i.e., $B_j(i)$ = Pr(N_j outputs v_i)
5. An initial state probability distribution π (i.e., $\pi(i)$ = probability that Si is the starting state.)

Forward Probability Calculation

Define $\alpha_t(i) = P(Q_t=S_i, O_1, \ldots, O_t)$

Given a sequence of length T, and an HMM with transition matrix A and output probability B,

1. Initialization
   $\alpha_1(i) = \pi(i) * B_i(o_1)$ for every I (i.e., every node S_i)
2. Iteration Step
   $\alpha_{t+1}(i) = (\sum_{j=1,N} \alpha_t(j)\, A_{j,i}) * B_i(o_{t+1})$ for every i

Backward Probability Calculation

Define $\beta_t(i) = P(O_{t+1}, \ldots, O_T \mid Q_t=S_i)$

Given a sequence of length T, and HMM as before,

1. Initialization
   $\beta_T(i) = 1$
2. Iteration
   $\beta_t(i) = \sum_j A_{i,j} * B_j(o_{t+1}) * \beta_{t+1}(j)$

time t. They all produce the same value. Here we'll use the one with the forward probabilities to produce a formula.

$$\text{Prob}(q_t = S_i \mid o_1 \ldots o_T) = \alpha_t(i) * \beta_t(i) \;/\; \sum_i \alpha_T(i)$$

The probability of being in any state at any time given the output R W B B is shown in Figure 5.

State\time	R (t= 1)	W (t = 2)	B (t = 3)	B (t = 4)
S1	.8	.54	.46	.44
S2	.2	.46	.54	.56

Figure 4: The probability of being in state S_i at time t, given output is R W B B

Note we can now do a weighted count over the corpus in order to re-estimate the output probabilities for the state. For instance, for state S1 we have a count of .8 for R, .54 for W, and .9 for B (since it occurred twice). These sum to 2.24, so we end up we P(R | S1) = .8 / 2.24 = .357. Figure 5 shows the reestimated output probabilities for all states and outputs.

State	R	W	B
S1	.357	.241	.401
S2	.114	.261	.625

Figure 5: The new output probabilities for the HMM

These probabilities can be computed directly from the forward and backward probabilities using the following standard re-estimation formula.

$$\text{new Prob}(w_k \mid S_i) = \frac{\sum_{t \text{ where } o_t=w_k} \alpha_t(i) * \beta_t(i)}{\sum_t \alpha_t(i) * \beta_t(i)}$$

Re-estimating the Transition Probabilities

To compute the probability of taking the transition $S_i \rightarrow S_j$, written as $A_{i,j}$ we can use a similar argument to the above. For a single time t, we use the forward probability to find out how likely we are to get to S_i at time t, then take $A_{i,j} * P(o_{t+1} \mid S_j)$ as the probability of taking the transition at time t, and then the backward probability to find out how likely we are to complete the observation sequence from state S_j, i.e.,

$$P(Q_t=S_i, Q_{t+1}=S_j \ \& \ o_1 \ldots o_T) = \alpha_t(i) * A_{i,j} * P(o_{t+1} \mid S_j) * \beta_{t+1}(j)$$

Thus, the probability of taking this transition at any time during the process is the sum of these terms over time t.

$$\sum_t \alpha_t(i) * A_{i,j} * P(o_{t+1} \mid S_j) * \beta_{t+1}(j)$$

To make this into a "count" we need to compute the probability of ever being in state S_i at any time t. We saw above that the probability of being in state S_i at time t is $\alpha_t(i) * \beta_t(i)$, so the probability for any time will be the sum over these

$$\sum_t \alpha_t(i) * \beta_t(i).$$

We can put these together to get the new estimate for the transition probability $A_{i,j}$

$$\text{New } A_{i,j} = (\sum_t \alpha_t(i) * A_{i,j} * P(o_{t+1} \mid S_j) * \beta_{t+1}(j)) / \sum_t \alpha_t(i) * \beta_t(i)$$

For instance, consider calculating the probability that we go from S1 to S2.

We need to consider this at each time step. For t=1 we have

$$\alpha_1(1) * A_{1,2} * P(B \mid S_2) * \beta_2(2)) = .24 * .4 * .3 * .09 = .0026$$

For t=2 we get .0024 and t=3 we get .0019. The sum of these gives .007. Likewise, if we add up the terms for S1-> S1 for all t, we get .0117. The denominator in each of these cases is equal to the sum of these two values (since it is all the cases that involve a transition from S1), so we get new transition estimates as follows:

$$\text{New } A_{1,1} = .007 / (.007+.0117) = .37$$

$$\text{New } A_{1,2} = .0117/(.007+.0117) = .63$$

The complete set of new transition probabilities is shown in Figure 6.

Transition	To S1	To S2
From S1	.63	.37
From S2	.31	.69

Figure 6: The new Transition probabilities.

We can see whether we have a better model now by looking at the probability of the corpus. We also have calculated the probability of the corpus under the old model:

$$P_{old}(RWBB) = \sum_i \alpha_T(i) = .01$$

We the new output and transition probabilities we get

$$P_{new}(RWBB) = .02$$

doubling the probability of the corpus. Another iteration brings the corpus probability to .028 and one more to .043!

At this stage, the new output and transition probabilities are shown in Figure 7.

Transition	To S1	To S2
From S1	.43	.57
From S2	.11	.89

New Transition Probabilities

	R	W	B
S1	.53	.27	.2
S2	.01	.23	.76

New Output Probabilities

Figure 7: The New HMM Probabilities After 4 Iterations

Training with a Corpus

In speech applications, we want to train our HMM models to optimize recognition performance over a wide number of observation sequences. But the method described above only considered a single sequence. We will consider a way to generalize this procedure by estimating the parameters for each of the instances in the training corpus and then combining them.

For instance, if we have four instances in our corpus, O1 = RWBB, O2 = RBWB, O3 = WRBR and O4 = RRBB, we compute good parameters for each. In the ideal model, the probability of each of these sequences should be equal, since each occurs the same number of times in the training corpus. So rather than combining the values computed for each on an equal basis, we weight each by the inverse of the probability assigned it by the current HMM. For example, say the current HMM assigns a probability of .3 to RWBB, RBWB and WRBR, and only .1 to RRBB. Then the weighting we use will be 1/.3 = 3.333 for the first three and 1/.1 = 10 for the last. By giving more weight to the low probability sequence, we make its parameter estimates more influential in the new estimate.

Given this, the reestimation formulas for a corpus of observation sequences $O^1, ..., O^k$ are:

$$A_{i,j} = \frac{\sum_{k=1,K} (\sum_t \alpha^k_t(i) * A^k_{i,j} * Prob(o_{t+1} \mid S_j) * \beta^k_t(i)/Prob(O^k))}{\sum_t \alpha^k_t(i) * \beta^k_t(i)/Prob(O^k)}$$

$\text{Prob}(o_i \mid S_j) = \sum_{k=1,N} \text{Prob}(o_i \mid S_j, Ok)$, for each observation o_i and state S_j.
where $\alpha^k_t(i)$ and $\beta^k_t(i)$ are the forward and backward probabilities for the the k'th observation sequence.

Given a corpus containing instances of M different words, we would perform this procedure on each set of instances for each word, producing an HMM for each word.