The Backward Algorithm

We have a hidden Markov model (HMM) with

N hidden states $\{q_1\ q_2 \ldots q_N\}$
M emission states $\{y_1, y_2, \ldots y_M\}$

And where

π_i is the initial probability for hidden state q_i
a_{ij} is the transition probability for moving from hidden state q_i to q_j
$b_i(y_j)$ is the emission probability for hidden state q_i and observation y_j

We are given a sequence of observed emission states $y_1, y_2, \ldots, y_t, \ldots y_T$

We have seen the forward algorithm, which involves calculating the probability of being in hidden state i after the first t observations (the forward probability). The backward probability describes the remaining observations from time $t+1$ to T assuming we are at hidden state i at time t.

$$\beta_t(i) = P(y_{t+1}, y_{t+2}, \ldots, y_T | q_t = i)$$

The algorithm is given:

Base case:

$$\beta_T(i) = 1, 1 \leq i \leq N$$

Recursion:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(y_{t+1}) \beta_{t+1}(j)\,, 1 \leq i \leq N, 1 \leq t < T$$

Termination:

$$P(y_1, y_2, \ldots, y_3) = \sum_{j=1}^{N} \pi_j\, b_j(y_1) \beta_1(j)$$

The intuition is much like that for the forward algorithm, but in the opposite direction. Where the forward algorithm has its base case at $t = 1$ and we build the forward probabilities α towards T, the backward algorithm has its base case at the end, $t = T$ and we build the backward probabilities β towards $t = 1$.

Notice that we can calculate the forward and backward probabilities at any time t. The use of these probabilities form the foundation for the Baum-Welch expectation maximization algorithm.

Exercises

Theory

1.1 Derive the backward algorithm. You can refer to the derivation of the forward algorithm for guidance.

Programming

2.1 Write pseudocode for the backward algorithm
2.2 Analyze the backward algorithm's time complexity
2.3 Implement the backward algorithm in a language of your choice

Bonus

3.0 Understand the Baum-Welch algorithm
3.1 Write the algorithm steps
3.2 What are the limitation of the Baum-Welch algorithm?