



Finding the top - k viterbi paths in HMM

I need to write an algorithm that finds the top-k viterbi paths in a HMM (using the regular viterbi algorithm to find the best path).

I think I probably need to save a list $V_{t,N}$ of size k for each state N that contains the top-K paths that end in state N, but I am not so sure how to keep track of that list.. any ideas? Thanks

[algorithm](#) [hidden-markov-models](#) [viterbi](#)

edited Apr 10 '13 at 18:09



[anil](#)

1,816 10 20

asked Dec 7 '12 at 17:47



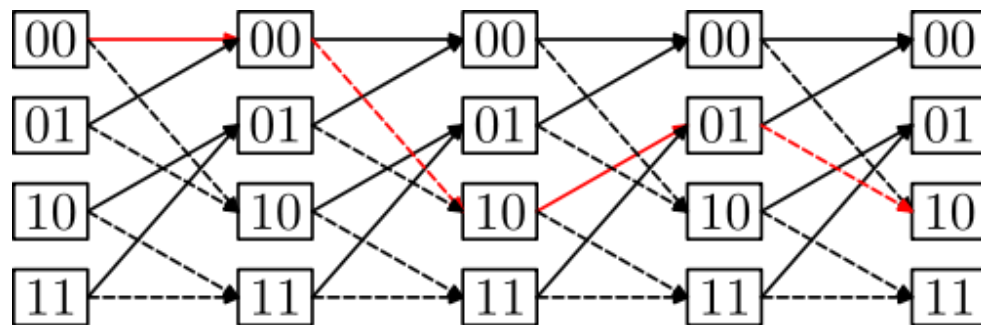
[John Smith](#)

80 2 5

1 You need an n-best decoder, which is typically accomplished with a beam search. – [bmargulies](#) Dec 7 '12 at 17:52

2 Answers

We can solve this with some care. It is easiest to see by looking at the trellis structure of hmm:



In this example the hidden states are 00, 01, 10, 11, denote the set of these four as S . The observations are not shown, but assume they are 0,1.

Suppose that we have the correct Transition Matrix:

```
transition[4][4]
```

Emission probabilities:

```
emissions[4][2]
```

And initial probabilities:

```
p[2]
```

So every column represents the hidden states, and the goal of Viterbi is to compute the most likely hidden state sequence given the observations. Now let $\alpha(i, t)$ = the largest probability that the hidden state sequence is in state i (i is one of 00, 01, 10, 11), at time t where the observation at time t is o_t (o_t is one of 0, 1). Let the first observation be denoted o_1 . It can be computed efficiently as:

```
alpha(i, 1) = p[i] * emissions[i][o_1]
alpha(i, t) = emissions[i][o_t] * max_{k in states} (alpha(k, t-1) * transition[k][i])
```

In order to find the best path, we keep pointers backwards in the $\alpha(i, t)$ step, to the state which maximized the max function in above. Finally we just examine all of the $\alpha(i, T)$ and for i in states, and find which one is largest, then follow it back to get the most likely state sequence.

Now we need to extend this to store top k -paths. Currently at each $\alpha(i, t)$ we only store one parent. However suppose we stored the top k predecessors. So each $\alpha(i, t)$ corresponds not only to a most likely value and the node which it transitioned from, but a list of the top k nodes it could have transitioned from and their values in sorted order.

This is easy to do, in that instead of doing max, and take only one preceding node we take the top k preceding nodes and store them. Now for the base case there is no preceding node so $\alpha(i, 1)$, still is only a single value. When we arrive at an arbitrary column (say t) and want to find the top- k paths ending at a node (i) in that column, we must find the top k predecessors to come from and the top paths to take from them.

This is as if we have the following problem, a matrix, m , of size 4 by k , where a row represents a preceding state and $m[\text{state}]$ represents the top k probabilities for paths ending there. Thus each row of m is sorted by largest to smallest, the problem now becomes find:

```
Best_K_Values(t, i) = Top K over all i,preceding_state,k (emissions[i][o_t] *  
m[preceding_state][k] * transition[preceding_state][i])
```

Now this looks daunting but take some time to understand it, we can solve the top k from sorted matrix problem using a heap in $O(4 \log k)$ or $O(\text{numStates} \log k)$ in general.

See this slight variation [Kth smallest element in sorted matrix](#), just note that in our case the columns aren't sorted, but the idea there still applies.

If you are still reading, then note that the overall complexity of this method is $O((\text{numStates} \log k) * \text{numStates} * t) = O(\text{numStates}^2 * t * \log k)$ (I believe that's correct complexity).

This may be hard to follow, but please let me know if you have any questions, or I have done something incorrectly.

edited Mar 21 '13 at 6:43

answered Mar 16 '13 at 20:26



anil

1,816 10 20

There's actually extensive literature on this. It took some digging, but I found this.

List Viterbi Decoding Algorithms with Applications

<http://www.ensc.sfu.ca/people/faculty/cavers/ENSC805/readings/42comm02-seshadri.pdf>

answered Dec 22 '14 at 4:08



Madmartigan

314 3 6