

The Viterbi Algorithm

Tandy Warnow

February 20, 2017

The Viterbi Algorithm

Tandy Warnow

Announcements

Wednesday, Feb 22: quiz on reading assignments up to this week.

Using profile HMMs in biomolecular sequence analysis

Profile HMMs have multiple uses, including:

1. Computing multiple sequence alignments (MSAs)
2. Classifying newly discovered proteins into protein families
3. Detecting remote homology
4. Inferring protein or RNA structure

Using profile HMMs to compute MSA

Input:

- ▶ set S of sequences, each from a known gene (e.g., all hemoglobin sequences).
- ▶ profile HMM for the gene

How can we align the sequences in S ?

Using profile HMMs to compute MSA

Answer:

- ▶ For each $s \in S$, find the maximum likelihood path through the profile HMM; this defines an alignment of s to the profile
- ▶ Align all the sequences using these paths

Notes:

- ▶ The match states for the profile HMM are the ones that represent columns that more than one sequence can contribute to
- ▶ The insertion states represent columns for individual sequences
- ▶ The alignment for sequence s is determined by the match states that it passes through in its maximum likelihood path

Computing the Maximum Probability Path

Problem formulation:

We are given a string $x = x_1x_2 \dots x_n \in \Sigma^n$ and a profile HMM called H , and we wish to find the path through H that has the highest probability of generating x .

We will do this indirectly, by computing the the maximum probability of *any* path through H that can generate x , and then use backtracking to find the actual path.

The algorithm that does this is called the **Viterbi Algorithm**.

Notation for profile HMMs

- Σ is the alphabet (e.g., all the symbols that can be generated by any state in H).
- a_{vw} is the transition probability for the directed edge vw (i.e., the probability of moving directly to w from v , given that you start at v).
- $e_v(b)$ is the probability of emitting symbol $b \in \Sigma$ when in state $v \in V$, under the assumption that v is not a deletion state.
- The string $x = x_1x_2 \dots x_n \in \Sigma^n$, and its i^{th} prefix $x_{1\dots i}$ is $x_1x_2 \dots x_i$; by convention, the 0^{th} prefix is the empty string.
- V will denote the set of states of H , and will include a start state v_0 and an end state v_L .

Computing the Maximum Probability Path

Problem formulation:

Let $MPP(v, i)$ denote the maximum probability of generating the prefix x_1, x_2, \dots, x_i for any path in H that begins at v_0 and ends at state v .

Computing the Maximum Probability Path

Problem formulation:

Recall $MPP(v, i)$ denotes the maximum probability of generating the prefix x_0, x_1, \dots, x_i for any path in H that begins at the state and ends at state v .

Hence, $MPP(v_L, n)$ is the maximum probability of generating x using any path through H , where n is the length of x .

Computing the Maximum Probability Path

Recall $MPP(v, i)$ denote the maximum probability of generating the prefix x_0, x_1, \dots, x_i for any path in H that begins at the state and ends at state v .

We will use dynamic programming to compute $MPP(v, i)$ for every math and insertion state in H and every index $0 \leq i \leq n$, where n is the length of x .

- ▶ What are the boundary conditions?
- ▶ How can we compute each subproblem?
- ▶ What order should we compute the subproblems in?
- ▶ If we want the path and not just the maximum probability, we need to do backtracking. But how?

Computing the Maximum Probability Path

The recursion:

$$MPP(v, i) = \begin{cases} 1 & v = v_0 \text{ and } i = 0 \\ 0 & v = v_0 \text{ and } i > 0 \\ \max_w \{MPP(w, i-1)a_{wv}e_v(x_i)\} & v \text{ is a match state or an insertion state} \\ \max_w \{MPP(w, i)a_{wv}\} & \text{otherwise} \end{cases}$$

Computing the Maximum Probability Path

The order in which we compute subproblems:

- ▶ Order the vertices so that each vertex v appears after every vertex w such that $w \rightarrow v$ is an arc in the HMM.
- ▶ Compute $MPP(v, i)$ after $MPP(w, j)$ for all w that precede v .
- ▶ If $v = w$ then compute $MPP(v, i)$ after $MPP(v, j)$ if $j < i$.

Running time: $O(|V|n)$, since each vertex in H has $O(1)$ neighbors.

Backtracking: how do we compute the path with the maximum probability of generating x ?

Summary

- ▶ Profile HMMs have multiple uses, including computing multiple sequence alignments
- ▶ The Viterbi algorithm finds the best path through a profile HMM for a sequence
- ▶ The Viterbi algorithm can be used to compute a multiple sequence alignment

Reminder

Reading assignment quiz on Wednesday covering

- ▶ All material from the textbook listed as reading assignment up through February 22
- ▶ All posted course presentations through February 20
- ▶ The three papers on genome assembly