1 Introduction

A hidden Markov model (HMM) is a tuple $(H, \Sigma, T, E, \mathbb{P})$, where $H = \{1, \ldots, |H|\}$ is the set of hidden states, Σ is the set of symbols, $T \subseteq H \times H$ is the set of transitions, $E \subseteq H \times \Sigma$ is the set of emissions, and $\mathbb{P}$ is the probability function for elements of T and E, satisfying the following conditions:

- There is a single *start state* $h_{\texttt{start}} \in H$ with no transitions $(h, h_{\texttt{start}}) \in T$, and no emissions (for this reason, $h_{\texttt{start}}$ is also called a *silent state*).
- There is a single *end state* $h_{\texttt{end}} \in H$ with no transitions $(h_{\texttt{end}}, h) \in T$, and no emissions (also $h_{\texttt{end}}$ is a *silent state).*
- Let $\mathbb{P}(h|h')$ denote the probability for the transition $(h, h') \in T$, and let $\mathbb{P}(c|h)$ denote the probability of an emission $(h, c) \in E$, for $h', h \in H$ and $c \in \Sigma$. It must hold that

  $$\sum_{h \in H} \mathbb{P}(h|h') = 1, \text{ for all } h' \in H \setminus \{h_{\texttt{end}}\}.$$

  Especially,

  $$\sum_{h \in H} \mathbb{P}(h|h_{\text{start}}) = 1.$$

  ($\mathbb{P}(h, h')$ gives the transition probability from state h' to h. Reverse order of the arc (h, h').)

Observe that we denote the probability of transition *from* h' *to* h by $\mathbb{P}(h|h')$, rather than with a notation like $p(h', h)$. "h given h'."

2 Definitions

A *path through an HMM* is a sequence P of hidden states $= P = p_0 p_1 p_2 \cdots p_n p_{n+1}$, where $(p_i, p_{i+1}) \in T$, for each $i \in \{0, \ldots, n\}$. The *joint probability* of P and a sequence $S = s_1 s_2 \cdots s_n$, with each $s_i \in \Sigma$, is

$$\mathbb{P}(P, S) = \prod_{i=0}^{n} \mathbb{P}(p_{i+1}|p_i) \prod_{i=1}^{n} \mathbb{P}(s_i|p_i).$$

We will be mainly interested in the set $\mathcal{P}(n)$ of all paths $p_0 p_1 \cdots p_{n+1}$ through the HMM, of length $n + 2$, such that $p_0 = h_{\texttt{start}}$ and $p_{n+1} = h_{\texttt{end}}$.

3 Problems

Given an HMM M over an alphabet Σ, and a sequence $S = s_1 s_2 \cdots s_n$, with each $s_i \in \Sigma$, find the path $P^\star$ in M having the highest probability of generating S, namely

$$P^\star = \arg\max_{P \in \mathcal{P}(n)} \mathbb{P}(P, S) = \arg\max_{P \in \mathcal{P}(n)} \prod_{i=0}^{n} \mathbb{P}(p_{i+1}, p_i) \prod_{i=1}^{n} \mathbb{P}(s_i, p_i). \tag{1}$$

Given an HMM M over an alphabet Σ, and a sequence $S = s_1 s_2 \cdots s_n$, with each $s_i \in \Sigma$, compute the probability

$$\mathbb{P}(S) = \sum_{P \in \mathcal{P}(n)} \mathbb{P}(P, S) = \sum_{P \in \mathcal{P}(n)} \prod_{i=0}^{n} \mathbb{P}(p_{i+1}, p_i) \prod_{i=1}^{n} \mathbb{P}(s_i | p_i). \tag{2}$$

For a path $P = p_0 p_1 p_2 \cdots p_n$ through the HMM, we define

$$\mathbb{P}_{\texttt{prefix}}(P, S) = \prod_{i=0}^{n-1} \mathbb{P}(p_{i+1} | p_i) \prod_{i=1}^{n} \mathbb{P}(s_i | p_i).$$

Given a path $P = p_1 p_2 \cdots p_n p_{n+1}$ through the HMM, we define

$$\mathbb{P}_{\texttt{suffix}}(P, S) = \prod_{i=1}^{n} \mathbb{P}(p_{i+1} | p_i) \prod_{i=1}^{n} \mathbb{P}(s_i | p_i).$$

4 The Viterbi algorithm

The Viterbi algorithm solves the Problem 1. For every $i \in \{1, \ldots, n\}$ and every $h \in \{1, \ldots, |H|\}$, define

$$v(i, h) = \max \left\{ \mathbb{P}_{\texttt{prefix}}(P, s_1 \cdots s_i) | P = h_{\texttt{start}} p_1 \cdots p_{i-1} h \right\}$$

as the largest probability of a path starting in state $h_{\texttt{start}}$ and ending in state h, given that the HMM generated the prefix $s_1 \cdots s_i$ of S (symbol s_i being emitted by state h).

We can easily derive the following recurrence relations for $v(i, h)$:

$$\begin{aligned} v(i, h) &= \max \left\{ \mathbb{P}_{\texttt{prefix}}(h_{\texttt{start}}, p_1 \cdots p_{i-1} h', s_1 \cdots s_{i-1}) \mathbb{P}(h | h') \mathbb{P}(s_i | h) | (h', h) \in T \right\} \\ &= \mathbb{P}(s_i, |h) \max\{ v(i-1, h') \mathbb{P}(h | h') | (h', h) \in T \}, \end{aligned} \tag{3}$$

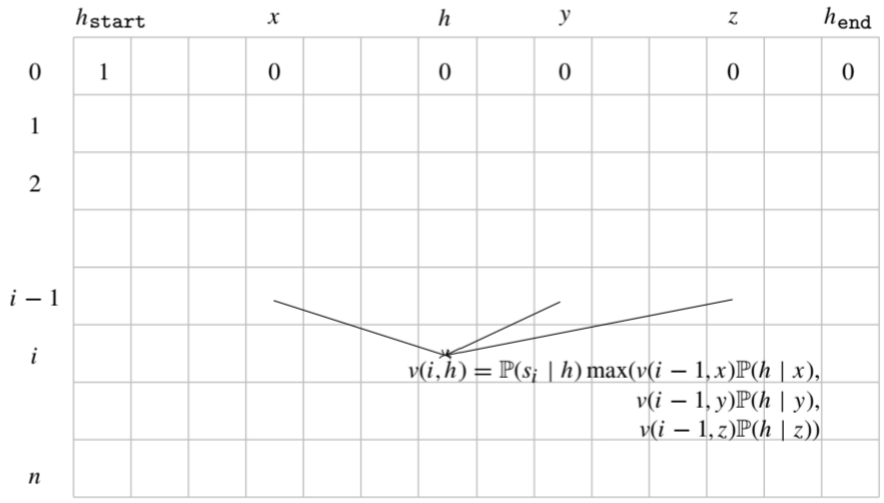

Figure 1: The idea behind the Viterbi algorithm, assuming that the predecessors of state h are the states x, y, and z.

,

where we take by convention $v(0, h_{\texttt{start}} = 1$ and $v(0, h) = 0$ for all $h \neq h_{\texttt{start}}$. Indeed, $v(i, h)$ equals the largest probability of getting to a predecessor h' of h, having generated the prefix sequence $s_1 \cdots s_{i-1}$, multiplied by the probability of the transition (h', h) and by $\mathbb{P}(s_i|h)$.

The largest probability of a path for the entire string S (that is, the value maximizing 1) is the largest probability of getting to a predecessor h' of $h_{\texttt{end}}$, having generated the entire sequence of $S = s_1 \cdots s_n$ (symbol s_n begin emitted by state h'), multiplied by the probability of the final transition $(h', h_{\texttt{end}}$. Expressed in terms of v,

$$\max_{P \in \mathcal{P}(n)} \mathbb{P}(P, S) = \max \left\{ v(n, h')\mathbb{P}(h_{\texttt{end}}|h')|(h', h_{\texttt{end}} \in T \right\}. \tag{4}$$

The values $v(\cdot, \cdot)$ can be computed by filling a table $V[0..n, 1..|H|]$ row-by-row in $O(n|T|)$ time. The most probable path can be traced back in the standard dynamic programming manner, by checking which predecessor h' of $h_{\texttt{end}}$ maximizes 4 and then, iteratively, which predecessor h' of the current state h maximizes **??**. Figure 1 illustrates the dynamic programming recurrence.

5 The forward and backward algorithms

The *forward* algorithm solves the Problem