
Notes Regarding Computations in OpenHTMM

Amit Gruber
The Hebrew University of Jerusalem
Israel

Ashok C. Popat
Google, Inc.
USA

Abstract

1 Introduction

In this note we explain how probabilistic inference on the Markov chain is implemented in OpenHTMM. In HTMM [1], the pairs (topic, ψ) form a hidden Markov chain if the parameters and the sequence of observed words are given. The E step of the EM inference requires the computation of posterior probabilities in this Markov chain. Finding the most probable sequence of topics along a document after we have finished training (i.e. we have estimated the parameters) using EM requires MAP estimation in the Markov chain.

The transition matrix in this Markov chain has a special form that depends on the parameters ϵ and θ . In this note we explain how to take advantage of this special case of a hidden Markov model for fast computation of two inference problems: (1) computation of the distribution over the hidden (state, ψ) pairs and (2) finding the most probable sequence of topics along the document.

The standard solution for the first problem is the Forward - Backward algorithm (a.k.a the Alpha Beta algorithm) and for the second problem is the Viterbi algorithm. Both require $O(TK^2)$ (where T is the length of the chain and K is the number of possible states). The solution described here requires only $O(TK)$ operations.

2 Definitions and standard computation of best path using Viterbi

Let $w_0^{T-1} = (w_0, \dots, w_{T-1})$ denote the sequence of T words observed in a particular document. We shall use the term "level" to refer to position in the sequence. We hypothesize a hidden sequence $z_0^{T-1} = (z_0, \dots, z_{T-1})$ of topics, each of which assumes a value in the set

$\{0, \dots, K-1\}$. The topic transitions obey a Markov model described by a transition matrix of a special form that contains only K free parameters, rather than the usual $K^2 - K$. In order to describe this restricted form, it is necessary to introduce an additional random variable $\psi_i \sim \text{Bern}(\epsilon)$ that governs whether or not the topic is re-drawn at random at level i . If $\psi_i = 0$, then $z_i = z_{i-1}$ deterministically; if $\psi_i = 1$, then $z_i = k$ with probability θ_k , $k = 0, \dots, K-1$, where θ is a fixed multinomial. Note that even if $\psi_i = 1$, it is possible that $z_i = z_{i-1}$ (with conditional probability $\theta_{z_{i-1}}$).

Rather than encode the topic directly, the state s_i in our model at level i comprises both z_i and ψ_i . We use the encoding $s_i = z_i + K(1 - \psi_i)$. Thus, the state s_i ranges over the $2K$ values $0, \dots, 2K-1$, where the first K values correspond to the topic being drawn anew, and the last K corresponding to the topic being retained from the previous level. The situation is most easily visualized using a *trellis* diagram (See Fig. 2).

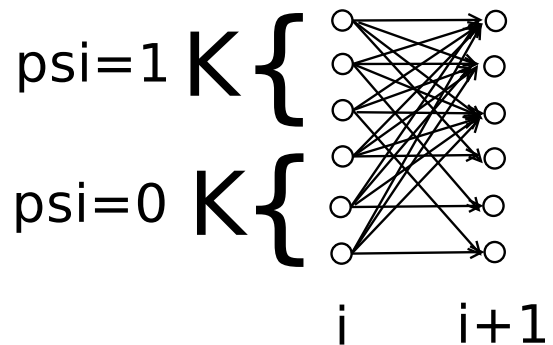


Figure 1: Trellis diagram for state sequences and associated variables. Each node at level i corresponds to one of the $2K$ possible values for the state s_i . TODO: make this figure neater and more complete.

Accordingly, the state sequence $s_0^{T-1} = (s_0, \dots, s_{T-1})$ is governed by a probability mass function π over the initial

state s_0 , along with the transition matrix $(a_{k,k'})$ having elements

$$a_{kk'} = \begin{cases} \epsilon\theta_{k'} & \text{if } 0 \leq k' < K \\ 1 - \epsilon & \text{if } K \leq k' < 2K, k \in \{k' - K, k'\} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Note that there are K free parameters in this definition of $(a_{k,k'})$: there are $K - 1$ free parameters in θ , plus one additional free parameter ϵ .

Each topic z has an associated topic-specific multinomial distribution $p'(w|z)$ over words w . For notational convenience we define

$$p(w|s) = \begin{cases} p'(w|z = s) & \text{if } 0 \leq s < K \\ p'(w|z = s - K) & \text{if } K \leq s < 2K \end{cases} \quad (2)$$

Using this set-up, the application of the Viterbi algorithm becomes standard. Treating the observed word sequence $w_0^{T-1} = (w_0, \dots, w_{T-1})$ as fixed, the probability distribution over the state sequence S_0^{T-1} is

$$\begin{aligned} Pr(s_0^{T-1} \mid w_0^{T-1}, \pi, a, p') \\ = \pi(s_0)p(w_0|s_0) \prod_{i=1}^{T-1} a_{s_{i-1}s} p(w_i|s) \end{aligned} \quad (3)$$

We wish to find a state sequence that maximizes this probability. To do so, we associate with each node (i, s) in the trellis the quantity $\delta(i, s)$ recursively defined as

$$\delta(i, s) = \max_{s'} [\delta(i-1, s') a_{s's} p(w_i|s)], \quad (4)$$

for $s = 0, \dots, 2K - 1, i = 1, \dots, T - 1$, where we take $\delta(0, s_0) = \pi(s_0)p(w_0|s_0)$. In addition, we associate with each node a *backpointer*

$$b(i, s) = \arg \max_{s'} [\delta(i-1, s') a_{s's} p(w_i|s)] \quad (5)$$

that records the previous state that yielded the maximum in Eq. 4. To find a state sequence that maximizes Eq. 3, we update $\delta(i, s)$ using Eq. 4 successively for $i = 1, \dots, T - 1$. The maximizing state sequence $(s_0^*, \dots, s_{T-1}^*)$ is then obtained by a *backtracking* pass $s_{i-1}^* = b_i(s_i^*), i = T - 2, \dots, 0$, where

$$s_{T-1}^* = \arg \max_s \delta(T-1, s)$$

2.1 Efficient Computation

In this subsection we explain how to take advantage of the special form of the transition matrix for fast inference in the Markov chain.

Standard computation of $\delta(i, s)$ in equation 4 requires $O(K)$ for each possible state s in each level i , hence the total run time of $O(TK^2)$.

We shall distinguish between two different kinds of nodes in the computation of δ : (1) nodes among the first K (i.e. a topic was drawn as a result of a lottery) and (2) nodes among the last K (no lottery took place, the topic was simply copied from the previous level).

In the first case, we could have come to node s in level i from any node of the $2K$ in level $i - 1$. However, the transition probabilities from each one of these $2K$ nodes to the node s in level i is *the same*. This means that we need only find the most probable node in level $i - 1$ and compute:

$$\delta(i, s) = [\max_{s'} \{\delta(i-1, s')\}] a_{*s} \cdot p(w_i|s), \quad (6)$$

where a_{*s} denotes the transition probability from any previous state to state s by lottery. Equation 6 shows that the computation of $\delta(i, s)$ decouples into the computation of $[\max_{s'} \{\delta(i-1, s')\}]$ and the computation of $a_{*s} \cdot p(w_i|s)$. The first term is independent of s , hence can be computed *only once for each level* in the cost of $O(K)$. The second term depends only on s and its computation requires $O(1)$. Hence the computation of $\delta(i, s)$ for all $1 \leq s \leq K$ requires $O(K)$.

In the second case, there are only two options: Node $K + s$ of the i could be approached only from node s or from node $K + s$ of the $i - 1$ level. Hence we simply need to select

$$\begin{aligned} \delta(i, s + K) = [\max_{s, s+K} \{\delta(i-1, s) a_{ss}, \delta(i-1, s+K) a_{s+Ks}\}] \\ \cdot p(w_i|s) \end{aligned} \quad (7)$$

Therefore the computation of $\delta(i, s + K)$ for all $1 \leq s \leq K$ requires $O(K)$ as well.

The total run time will be $O(TK)$.

2.2 Computing Marginal Probabilities

The Estep of the EM algorithm consists of computing posterior probabilities in the Markov chain. This is done using a special version of the Forward Backward algorithm, taking advantage of the special form of the transition matrix. The basic idea is very similar to the efficient implementation of the Viterbi algorithm (one difference is that a forward pass is required).

[This section may be expanded in future]

References

- [1] GRUBER, A., ROSEN-ZVI, M., AND WEISS, Y. Hidden topic markov models. In *Proceedings of Artificial Intelligence and Statistics (AISTATS)* (San Juan, Puerto Rico, March 2007).