The Forward-Backward Algorithm

Michael Collins

1 Introduction

This note describes the *forward-backward algorithm*. The forward-backward algorithm has very important applications to both hidden Markov models (HMMs) and conditional random fields (CRFs). It is a dynamic programming algorithm, and is closely related to the Viterbi algorithm for decoding with HMMs or CRFs.

This note describes the algorithm at a level of abstraction that applies to both HMMs and CRFs. We will also describe its specific application to these cases.

2 The Forward-Backward Algorithm

The problem set-up is as follows. Assume that we have some sequence length m, and some set of possible states $\mathcal{S}$. For any state sequence $s_1 \ldots s_m$ where each $s_i \in \mathcal{S}$, we define the *potential* for the sequence as

$$\psi(s_1 \ldots s_m) = \prod_{j=1}^{m} \psi(s_{j-1}, s_j, j)$$

Here we define s_0 to be *, where * is a special start symbol in the model. Here $\psi(s, s', j) \geq 0$ for $s, s' \in \mathcal{S}$, $j \in \{1 \ldots m\}$ is a potential function, which returns a value for the state transition s to s' at position j in the sequence.

The potential functions $\psi(s_{j-1}, s_j, j)$ might be defined in various ways. As one example, consider an HMM applied to an input sentence $x_1 \ldots x_m$. If we define

$$\psi(s', s, j) = t(s|s')e(x_j|s)$$

then

$$\begin{aligned}
\psi(s_1 \ldots s_m) &= \prod_{j=1}^{m} \psi(s_{j-1}, s_j, j) \\
&= \prod_{j=1}^{m} t(s_j|s_{j-1})e(x_j|s_j)
\end{aligned}$$

$$= \quad p(x_1 \dots x_m, s_1 \dots s_m)$$

where $p(x_1 \dots x_m, s_1 \dots s_m)$ is the probability mass function under the HMM.

As another example, consider a CRF where we have a feature-vector definition $\underline{\phi}(x_1 \dots x_m, s', s, j) \in \mathbb{R}^d$, and a parameter vector $\underline{w} \in \mathbb{R}^d$. Assume again that we have an input sentence $x_1 \dots x_m$. If we define

$$\psi(s', s, j) = \exp\left(\underline{w} \cdot \underline{\phi}(x_1 \dots x_m, s', s, j)\right)$$

then

$$\begin{aligned}
\psi(s_1 \dots s_m) &= \prod_{j=1}^{m} \psi(s_{j-1}, s_j, j) \\
&= \prod_{j=1}^{m} \exp\left(\underline{w} \cdot \underline{\phi}(x_1 \dots x_m, s_{j-1}, s_j, j)\right) \\
&= \exp\left(\sum_{j=1}^{m} \underline{w} \cdot \underline{\phi}(x_1 \dots x_m, s_{j-1}, s_j, j)\right)
\end{aligned}$$

Note in particular, by the model form for CRFs, it follows that

$$p(s_1 \dots s_m | x_1 \dots x_m) = \frac{\psi(s_1 \dots s_m)}{\sum_{s_1 \dots s_m} \psi(s_1 \dots s_m)}$$

The forward-backward algorithm is shown in figure 1. Given inputs consisting of a sequence length m, a set of possible states $\mathcal{S}$, and potential functions $\psi(s', s, j)$ for $s, s' \in \mathcal{S}$, and $j \in \{1 \dots m\}$, it computes the following quantities:

1. $Z = \sum_{s_1 \dots s_m} \psi(s_1 \dots s_m)$.
2. For all $j \in \{1 \dots m\}$, $a \in \mathcal{S}$,

$$\mu(j, a) = \sum_{s_1 \dots s_m : s_j = a} \psi(s_1 \dots s_m)$$

3. For all $j \in \{1 \dots (m-1)\}$, $a, b \in \mathcal{S}$,

$$\mu(j, a, b) = \sum_{s_1 \dots s_m : s_j = a, s_{j+1} = b} \psi(s_1 \dots s_m)$$

Inputs: Length m, set of possible states $\mathcal{S}$, function $\psi(s, s', j)$. Define * to be a special initial state.

Initialization (forward terms): For all $s \in \mathcal{S}$,

$$\alpha(1, s) = \psi(*, s, 1)$$

Recursion (forward terms): For all $j \in \{2 \dots m\}, s \in \mathcal{S}$,

$$\alpha(j, s) = \sum_{s' \in \mathcal{S}} \alpha(j-1, s') \times \psi(s', s, j)$$

Initialization (backward terms): For all $s \in \mathcal{S}$,

$$\beta(m, s) = 1$$

Recursion (backward terms): For all $j \in \{1 \dots (m-1)\}, s \in \mathcal{S}$,

$$\beta(j, s) = \sum_{s' \in \mathcal{S}} \beta(j+1, s') \times \psi(s, s', j+1)$$

Calculations:

$$Z = \sum_{s \in \mathcal{S}} \alpha(m, s)$$

For all $j \in \{1 \dots m\}, a \in \mathcal{S}$,

$$\mu(j, a) = \alpha(j, a) \times \beta(j, a)$$

For all $j \in \{1 \dots (m-1)\}, a, b \in \mathcal{S}$,

$$\mu(j, a, b) = \alpha(j, a) \times \psi(a, b, j+1) \times \beta(j+1, b)$$

Figure 1: The forward-backward algorithm.

3 Application to CRFs

The quantities computed by the forward-backward algorithm play a central role in CRFs. First, consider the problem of calculating the conditional probability

$$p(s_1 \ldots s_m | x_1 \ldots x_m) = \frac{\exp\left(\sum_{j=1}^{m} \underline{w} \cdot \underline{\phi}(x_1 \ldots x_m, s_{j-1}, s_j, j)\right)}{\sum_{s_1 \ldots s_m} \exp\{\left(\sum_{j=1}^{m} \underline{w} \cdot \underline{\phi}(x_1 \ldots x_m, s_{j-1}, s_j, j)\right)}$$

The numerator in the above expression is easy to compute; the denominator is more challenging, because it requires a sum over an exponential number of state sequences. However, if we define

$$\psi(s', s, j) = \exp\left(\underline{w} \cdot \underline{\phi}(x_1 \ldots x_m, s', s, j)\right)$$

in the algorithm in figure 1, then as we argued before we have

$$\psi(s_1 \ldots s_m) = \exp\left(\sum_{j=1}^{m} \underline{w} \cdot \underline{\phi}(x_1 \ldots x_m, s_{j-1}, s_j, j)\right)$$

It follows that the quantity Z calculated by the algorithm is equal to the denominator in the above expression; that is,

$$Z = \sum_{s_1 \ldots s_m} \exp\left(\sum_{j=1}^{m} \underline{w} \cdot \underline{\phi}(x_1 \ldots x_m, s_{j-1}, s_j, j)\right)$$

Next, recall that the key difficulty in the calculation of the gradient of the log-likelihood function in CRFs was to calculate the terms

$$q_j^i(a, b) = \sum_{\underline{s}: s_{j-1}=a, s_j=b} p(\underline{s} | \underline{x}^i; \underline{w})$$

for a given input sequence $\underline{x}^i = x_1^i \ldots x_m^i$, for each $j \in \{2 \ldots m\}$, for each $a, b \in \mathcal{S}$ (see the note on log-linear models). Again, if we define

$$\psi(s', s, j) = \exp\left(\underline{w} \cdot \underline{\phi}(x_1^i \ldots x_m^i, s', s, j)\right)$$

then it can be verified that

$$q_j^i(a, b) = \frac{\mu(j, a, b)}{Z}$$

where $\mu(j, a, b)$ and Z are the terms computed by the algorithm in figure 1.