CS 4650/7650, Lecture 12
Learning in conditional random fields

Jacob Eisenstein

September 26, 2013

0.1 Learning in CRFs

As with logistic regression, we need to learn weights to maximize the conditional log probability,

$$\begin{aligned}\ell =& \sum_i^{\#\text{instances}} \log P(\boldsymbol{y}_i|\boldsymbol{x}_i),\\ =& \sum_i \boldsymbol{w}^\top \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{y}_i) - \log \sum_{\boldsymbol{y}' \in \mathcal{Y}(\boldsymbol{x}_i)} \exp\left(\boldsymbol{w}^\top \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{y}')\right)\end{aligned}$$

And as in logistic regression, the derivative is a difference between observed and expected counts:

$$\begin{aligned}\frac{d\ell}{dw_j} =& \sum_i \text{count}(\boldsymbol{x}_i, \boldsymbol{y}_i)_j - E_{\boldsymbol{y}|\boldsymbol{x}_i;\boldsymbol{w}}[\text{count}(\boldsymbol{x}_i, \boldsymbol{y})_j]\\ \text{count}(\boldsymbol{x}_i, \boldsymbol{y}_i)_j =& \sum_n^N f_{n,j}(\boldsymbol{x}_i, y_{i,n}, y_{i,n-1}, n)\end{aligned}$$

For example:

- If feature j is $\langle CC, DT\rangle$, then $c_j(\boldsymbol{x}_n, \boldsymbol{y}_n)$ is the count of times DT follows CC in the sequence $\boldsymbol{y}_n$.
- If feature j is $\langle M : \text{-}thy, \text{JJ}\rangle$, then $\text{count}(\boldsymbol{x}_n, \boldsymbol{y}_n)_j$ is the count of words ending in *-thy* in $\boldsymbol{x}_n$ that are tagged JJ.

The expected feature counts are more complex.

- $E_{\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}}[\text{count}(\boldsymbol{x}_i, \boldsymbol{y})_j] = \sum_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x}_i)} P(\boldsymbol{y}|\boldsymbol{x}_i; \boldsymbol{w}) f_j(\boldsymbol{x}, \boldsymbol{y})$
- This looks bad: we have to sum over an exponential number of labelings again.
- But remember that the feature function decomposes $f_j(\boldsymbol{x}, \boldsymbol{y}) = \sum_n f_j(\boldsymbol{x}, y_n, y_{n-1}, n)$.

$$
\begin{aligned}
E_{\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}}[\text{count}(\boldsymbol{x}, \boldsymbol{y})_j] &= \sum_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x})} P(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{w}) f_j(\boldsymbol{x}, \boldsymbol{y}) \\
&= \sum_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x})} P(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{w}) \sum_{n}^{N} f_j(\boldsymbol{x}, y_{n,i}, y_{n,i-1}, n) \\
&= \sum_{n}^{N} \sum_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x})} P(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{w}) f_j(\boldsymbol{x}, y_{n,i}, y_{n,i-1}, n) \\
&= \sum_{n}^{N} \sum_{j,k\in\mathcal{Y}} \sum_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x}):y_{n-1}=j,y_n=k} P(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{w}) f_j(\boldsymbol{x}, y_{n,i}, y_{n,i-1}, n) \\
&= \sum_{n}^{N} \sum_{j,k\in\mathcal{Y}} f_j(\boldsymbol{x}, y_{n,i}, y_{n,i-1}, n) \sum_{\boldsymbol{y}\in\mathcal{Y}(\boldsymbol{x}):y_{n-1}=j,y_n=k} P(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{w}) \\
&= \sum_{n}^{N} \sum_{j,k\in\mathcal{Y}} f_j(\boldsymbol{x}, y_{n,i}, y_{n,i-1}, n) P(y_{n-1} = j, y_n = k|\boldsymbol{x}; \boldsymbol{w})
\end{aligned}
$$

- The expected feature counts can be computed efficiently if we know the **marginal** probabilities $P(y_n, y_{n-1}|\boldsymbol{x}; \boldsymbol{w})$.
- This is the probability of traversing the edge $y_{n-1} \to y_n$, conditioned on the entire observation $\boldsymbol{x}_{1:N}$. [Draw this in trellis]
- To compute this marginal probability, we will apply the forward-backward algorithm.

1 The forward-backward algorithm

Here we require the marginal probability, e.g. $P(y_n = \text{NNP}, y_{n-1} = \text{DET}|\boldsymbol{x}_{1:N})$.

That is, out of all possible taggings for $\boldsymbol{x}_{1:N}$, what is the probability of having y_n = NNP and y_{n-1} = DET? The forward-backward algorithm allows us to compute this probability efficiently.

Let's begin by rewriting

$$
\begin{aligned}
P(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{w}) =& \frac{\exp \boldsymbol{w}^\top \boldsymbol{f}(\boldsymbol{y},\boldsymbol{x})}{\sum_{\boldsymbol{y}'} \exp \boldsymbol{w}^\top \boldsymbol{f}(\boldsymbol{y}',\boldsymbol{x})} && (1)\\
Z(\boldsymbol{x},\boldsymbol{w}) =& \sum_{\boldsymbol{y}'} \exp \boldsymbol{w}^\top \boldsymbol{f}(\boldsymbol{y}',\boldsymbol{x}) && (2)\\
=& \frac{1}{Z(\boldsymbol{w},\boldsymbol{x})} \exp \boldsymbol{w}^\top \boldsymbol{f}(\boldsymbol{y},\boldsymbol{x}) && (3)\\
=& \frac{1}{Z(\boldsymbol{w},\boldsymbol{x})} \prod_n \exp \boldsymbol{w}^\top \boldsymbol{f}(\boldsymbol{x}, y_n, y_{n-1}, n) && (4)\\
=& \frac{1}{Z(\boldsymbol{w},\boldsymbol{x})} \prod_n \psi(y_n, y_{n-1}, n) && (5)\\
Z(\boldsymbol{x},\boldsymbol{w}) =& \sum_{\boldsymbol{y}'} \prod_n \psi(y'_n, y'_{n-1}, n) && (6)\\
& && (7)
\end{aligned}
$$

Now, for any tag k, there are a number of sequences that end in $y_n = k$. We'll define the sum of their scores as $\alpha_n(k) = \sum_{\boldsymbol{y}_{1:n}:y_n=k} \prod_m^n \psi(y_m, y_{m-1}, m)$. We can apply the forward algorithm to define α recursively:

$$
\begin{aligned}
\alpha_1(k) =& \psi(k, \star, 1) && (8)\\
\alpha_n(j) =& \sum_j \psi(k, j, n)\alpha_{n-1}(j). && (9)
\end{aligned}
$$

This allows us to compute $Z(\boldsymbol{x},\boldsymbol{w}) = \sum_k \alpha_N(k)$.

Now, we are interested in the marginal probability of all paths that pass through $y_{n-1} = j, y_n = k$,

$$P(y_{n-1} = j, y_n = k|\boldsymbol{x}) \propto \sum_{\boldsymbol{y}: y_{n-1}=j, y_n=k} \prod_m^N \psi(y_m, y_{m-1}, m) \tag{10}$$

$$P(y_{n-1} = j, y_n = k|\boldsymbol{x}) \propto \left(\sum_{\boldsymbol{y}_{1:n-1}: y_{n-1}=j} \prod_m^{n-1} \psi(y_m, y_{m-1}, m) \right) \tag{11}$$

$$\times \psi(k, j, n) \tag{12}$$

$$\times \left(\sum_{\boldsymbol{y}_{n+1:N}} \prod_m^N \psi(y_m, y_{m-1}, m) \right) \tag{13}$$

The first part of the product represents the sum over paths up to $y_{n-1} = j$. We already know how to compute this, it's the forward score:

$$\alpha_{n-1}(j) = \sum_{\boldsymbol{y}_{1:n-1}: y_{n-1}=j} \prod_m^{n-1} \psi(y_m, y_{m-1}, m) \tag{14}$$

The last part of the product represents the sum over label paths from y_{n+1} to y_N, given that $y_n = k$. This is the backwards score, and we can compute it recursively too.

$$\beta_n(k) = \sum_{\boldsymbol{y}_{n+1:N}} \prod_{m=n+1}^N \psi(y_m, y_{m-1}, m) \tag{15}$$

$$\beta_n(k) = \sum_{y_{n+1}} \psi(y_{n+1}, k, n+1) \sum_{\boldsymbol{y}_{n+2:N}} \prod_{m=n+2}^N \psi(y_m, y_{m-1}, m) \tag{16}$$

$$\beta_n(k) = \sum_{y_{n+1}} \psi(y_{n+1}, k, n+1) \beta_{n+1}(y_{n+1}) \tag{17}$$

$$\beta_N(k) = 1, \qquad \forall k \tag{18}$$

Therefore, we obtain the desired marginal probability

$$P(y_n = k, y_{n-1} = j|\boldsymbol{x}) = \alpha_{n-1}(j)\psi(k, j, n)\beta_n(k) / \sum_{k'} \alpha_N(k') \tag{19}$$

$$E_{\boldsymbol{y}|\boldsymbol{x}}[\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y})] = \frac{1}{\sum_{k'} \alpha_N(k')} \sum_{n}^{N} \sum_{j,k} \alpha_{n-1}(j)\psi(k, j, n)\beta_n(k)\boldsymbol{f}(\boldsymbol{x}, y_n = k, y_{n-1} = j, n) \tag{20}$$

We can also compute the marginal probability of an individual tag,

$$P(y_n = k|\boldsymbol{x}) = \frac{1}{\sum_{k'} \alpha_N(k')} \alpha_n(k)\beta_n(k) \tag{21}$$

2 Application to unsupervised HMMs

In an HMM, we have

$$P(\boldsymbol{x}, \boldsymbol{y}) = \prod_n P(x_n|y_n)P(y_n|y_{n-1}), \tag{22}$$

which we can relate to the notation above by setting

$$\psi(y_n, y_{n-1}, n) = P(x_n|y_n; \phi)P(y_n|y_{n-1}; \theta) \tag{23}$$

This means that

$$\alpha_n(k) = \sum_{\boldsymbol{y}_{1:n}: y_n = k} \prod_{n}^{m} \psi(y_m, y_{m-1}, m) \tag{24}$$

$$= \sum_{\boldsymbol{y}_{1:n}: y_n = k} \prod_{m} P(x_m, y_m|y_{m-1}) \tag{25}$$

$$= P(\boldsymbol{x}_{1:n}, y_n = k), \tag{26}$$

and

$$\beta_n(k) = \sum_{\boldsymbol{y}_{n:N}: y_n = k} \prod_{m=n}^{N} \psi(y_{n+1}, y_n, n) \tag{27}$$

$$= \sum_{\boldsymbol{y}_{n:N}: y_n = k} \prod_{m=n}^{N} P(x_{n+1}, y_{n+1}|y_n) \tag{28}$$

$$= P(\boldsymbol{x}_{n+1:N}|y_n) \tag{29}$$

Why is this useful? Suppose we want to do **unsupervised** part-of-speech tagging. We could use Expectation Maximization (EM) to estimate the parameters θ and ϕ. As usual, we replace the relative frequency estimate (from supervised learning) with its expectation,

$$\theta_{k,j} = \frac{E[\text{count}(y_n = k, y_{n-1} = j)]}{E[\text{count}(y_{n-1} = j)]} \tag{30}$$

We can compute these expectations using forward-backward:

$$\begin{aligned}
E[\text{count}(y_n = k, x_n = i)] &= \sum_n P(y_n = k|\boldsymbol{x}_{1:N})\delta(x_n = i) && (31)\\
&= \sum_n \frac{P(y_n = k, \boldsymbol{x}_{1:N})}{P(\boldsymbol{x}_{1:N})}\delta(x_n = i) && (32)\\
&= \sum_n \frac{P(y_n = k, \boldsymbol{x}_{1:n})P(\boldsymbol{x}_{n+1:N}|y_n = k)}{\sum_j P(y_N = j, \boldsymbol{x}_{1:N})}\delta(x_n = i) && (33)\\
&= \sum_n \frac{\alpha_n(k)\beta_n(k)}{\sum_j \alpha_N(j)}\delta(x_n = i) && (34)\\
E[\text{count}(y_n = k, y_{n-1} = j)] &= \sum_n P(y_n = k, y_{n-1} = j|\boldsymbol{x}_{1:N}) && (35)\\
&= \sum_n \frac{P(y_n = k, y_{n-1} = j, \boldsymbol{x}_{1:N})}{P(\boldsymbol{x}_{1:N})} && (36)\\
&\sum_n \frac{P(y_{n-1} = j, \boldsymbol{x}_{1:n-1})P(x_n, y_n|y_{n-1})P(\boldsymbol{x}_{n+1:N}|y_n = k)}{\sum_j P(y_N = j, \boldsymbol{x}_{1:N})} && (37)\\
&\sum_n \frac{\alpha_{n-1}(j)P(x_n|y_n = k)P(y_n = k|y_{n-1} = j)\beta_n(k)}{\sum_j \alpha_N(j)} && (38)
\end{aligned}$$