Learning Parameters for a Max-antecedent Co-reference Resolution Model

Jason D. M. Rennie
jrennie@csail.mit.edu

December 1, 2004

Define a similarity function on example pairs,

$$s(x_i, x_j; \vec{w}) = \vec{w} \cdot \vec{f}(x_i, x_j). \tag{1}$$

Define, for each example index $i \in \{1, \ldots, n\}$, for each possible antecedent label, $y \in y^i$, for each possible parameter vector, the index of the maximum similarity prior example,

$$j(i, y, \vec{w}) = \arg \max_{j<i|y_j=y} s(x_i, x_j; \vec{w}). \tag{2}$$

Define the set of unique labels of previous examples:

$$y^i = \{y|y \in \{y_1, y_2, \ldots, y_{i-1}\}\}, \quad \text{for } i \in \{1, \ldots, n\}. \tag{3}$$

Note that $y^1 = \emptyset$. Given a set of examples with labels, $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, the log-likelihood that our model assigns to the data is

$$L(\vec{w}) = \sum_i \log Q_i(\vec{w}) - \log Z_i(\vec{w}), \tag{4}$$

where

$$Q_i(\vec{w}) = \left\{ \begin{array}{cc} \exp\big(s(x_i, x_{j(i,y_i,\vec{w})}; \vec{w})\big) & \text{if } y_i \in y^i \\ 1 & \text{if } y_i = i \\ 0 & \text{othw.} \end{array} \right. , \tag{5}$$

and

$$Z_i(\vec{w}) = 1 + \sum_{y \in y^i} \exp(s(x_i, x_{j(i,y_i,\vec{w})}; \vec{w})). \tag{6}$$

Consider a procedure to learn a parameter vector for this model that runs in rounds. Let $\vec{w}^{(t)}$ be the parameter vector after the t^{th} iteration. Consider the "local" objective around $\vec{w}^{(t)}$,

$$L^{(t)}(\vec{w}) = \sum_i \log Q_i^{(t)}(\vec{w}) - \log Z_i^{(t)}(\vec{w}), \tag{7}$$

where

$$Q_i^{(t)}(\vec{w}) = \left\{ \begin{array}{cc} \exp\left(s(x_i, x_{j(i,y_i,\vec{w}^{(t)})}; \vec{w})\right) & \text{if } y_i \in y^i \\ 1 & \text{if } y_i = i \\ 0 & \text{othw.} \end{array} \right. , \tag{8}$$

and

$$Z_i^{(t)}(\vec{w}) = 1 + \sum_{y \in y^i} \exp(s(x_i, x_{j(i,y_i,\vec{w}^{(t)})}; \vec{w})). \tag{9}$$

Note that we have simply replaced $\vec{w}$ with $\vec{w}^{(t)}$ in the argument to $j(\cdot)$. $L^{(t)}$ is a convex function of $\vec{w}$. And, over the small region where $j(i,y,\vec{w}) = j(i,y,\vec{w}^{(t)})$, this "local" objective is equivalent to the model log-likelihood, $L^{(t)}(\vec{w}) \equiv L(\vec{w})$. This observation motivates the following simple, iterative optimization algorithm. Let $\vec{w}^{(t)}$ be the current parameter setting. Find the distance in the direction of the gradient that can be traveled before the "local" objective no longer aligns with the model log-likelihood. i.e. we want to find the distance along the gradient where there is a transition to a new "local" objective. Formally, we want to find

$$\alpha^* = \arg \inf_{\{\alpha | \vec{L}^{(t)}(\vec{w}^{(t)} + \alpha \nabla \vec{L}^{(t)}) \neq L(\vec{w}^{(t)} + \alpha \nabla \vec{L}^{(t)})\}} \alpha. \tag{10}$$

Since $L(\cdot)$ is entirely dependent on $j(\cdot)$, we can equivalently write this as

$$\alpha^* = \arg \inf_{\{\alpha | \exists i,y \text{ s.t. } j(i,y,\vec{w}^{(t)} + \alpha \nabla L^{(t)}) \neq j(i,y,\vec{w}^{(t)})\}} \alpha. \tag{11}$$

A necessary condition for $j(i,y,\vec{w}) \neq j(i,y,\vec{v})$ is that for some i and for some $y \in y^i$,

$$s(x_i, x_{j(i,y,\vec{w})}; \vec{w}) \geq s(x_i, x_{j(i,y,\vec{v})}; \vec{w}), \quad \text{and} \tag{12}$$

$$s(x_i, x_{j(i,y,\vec{w})}; \vec{v}) \leq s(x_i, x_{j(i,y,\vec{v})}; \vec{v}). \tag{13}$$

So, instead of searching over values of α, we search over possible values of $j(i,y,\vec{w})$, for each i, $y \in y^i$. Let k be such that $k < i$, $y_k = y$ and $k \neq j(i,y,\vec{w}^{(t)})$. Define $\Delta f(i,y,k,\vec{w}^{(t)}) = f(x_i, x_k; \vec{w}^{(t)}) - f(x_i, x_{j(i,y,\vec{w}^{(t)})})$. If there is some α such that $j(i,y,\vec{w} + \alpha \nabla \vec{L}^{(t)}) = k$, then the smallest such α is

$$\hat{\alpha}(i,y,k,\vec{w}^{(t)}) = -\frac{\vec{w}^{(t)} \cdot \Delta f(i,y,k,\vec{w}^{(t)})}{\nabla \vec{L}^{(t)} \cdot \Delta f(i,y,k,\vec{w}^{(t)})} \tag{14}$$

Note that $\hat{\alpha}$ will be negative if no such α exists. To find α^*, we simply take the smallest $\hat{\alpha}$ over allowable values of i, y, and k ($i \in \{1, \ldots, n\}$, $y \in y^i$, $k < i$, and $k \neq j(i,y,\vec{w}^{(t)})$).

Note that it is possible that $\alpha^* = 0$. In this case, we must use a direction other than the gradient[1]. Let $\hat{i}$, $\hat{y}$ and $\hat{k}$ be such that $\hat{\alpha} = 0$. Our new direction

[1] A "hack" to avoid such cases is to "overshoot" the iteration update. i.e. use the update $\vec{w}^{(t+1)} = \vec{w}^{(t)} + 2\alpha^* \nabla \vec{L}^{(t)}$. This avoids the "boundary" and may work well in practice, but there are no guarantees that such a procedure would converge.

must be such that $s(x_i, x_k; \vec{w}^{(t)}) = s(x_i, x_{j(i,y,\vec{w}^{(t)})}; \vec{w}^{(t)})$. We add this constraint and remove $\hat{\alpha}(\hat{i}, \hat{y}, \hat{k}, \vec{w}^{(t)})$ from consideration. If still $\alpha^* = 0$, we repeat, adding an additional constraint to the direction and removing the corresponding $\hat{\alpha}$ from consideration.

It can be shown that the regions are convex and hence have linear boundaries (since any non-linear boundary gurantees that either the inside or the outside is non-convex).

Note that if the number of regions is small, we can simply enumerate them and find the local maximum for each. The maximum of the local maximums is the global maximum.

If the number of regions is large, we can do random restarts to improve our chances of finding a point with objective value near the global max. And, if we keep track of the path of region traversals in each restart, we may terminate many of the restarts early. If a restart enters a region that had been part of a previous path, there is no need to continue (assuming we move to the max point in a region when we enter it).

We thank John Barnett for valuable comments and discussion.