Efficient Computation of Entropy Gradient for Semi-Supervised Conditional Random Fields

Gideon S. Mann and **Andrew McCallum**
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
gideon.mann@gmail.com, mccallum@cs.umass.edu

Abstract

Entropy regularization is a straightforward and successful method of semi-supervised learning that augments the traditional conditional likelihood objective function with an additional term that aims to minimize the predicted label entropy on unlabeled data. It has previously been demonstrated to provide positive results in linear-chain CRFs, but the published method for calculating the entropy gradient requires significantly more computation than supervised CRF training. This paper presents a new derivation and dynamic program for calculating the entropy gradient that is significantly more efficient—having the same asymptotic time complexity as supervised CRF training. We also present efficient generalizations of this method for calculating the label entropy of all sub-sequences, which is useful for active learning, among other applications.

1 Introduction

Semi-supervised learning is of growing importance in machine learning and NLP (Zhu, 2005). Conditional random fields (CRFs) (Lafferty et al., 2001) are an appealing target for semi-supervised learning because they achieve state-of-the-art performance across a broad spectrum of sequence labeling tasks, and yet, like many other machine learning methods, training them by supervised learning typically requires large annotated data sets.

Entropy regularization (ER) is a method of semi-supervised learning first proposed for classification tasks (Grandvalet and Bengio, 2004). In addition to maximizing conditional likelihood of the available labels, ER also aims to minimize the entropy of the *predicted* label distribution on unlabeled data. By insisting on peaked, confident predictions, ER guides the decision boundary away from dense regions of input space. It is simple and compelling—no pre-clustering, no "auxiliary functions," tuning of only one meta-parameter and it is discriminative.

Jiao et al. (2006) apply this method to linear-chain CRFs and demonstrate encouraging accuracy improvements on a gene-name-tagging task. However, the method they present for calculating the gradient of the entropy takes substantially greater time than the traditional supervised-only gradient. Whereas supervised training requires only classic forward/backward, taking time $O(ns^2)$ (sequence length times the square of the number of labels), their training method takes $O(n^2s^3)$—a factor of $O(ns)$ more. This greatly reduces the practicality of using large amounts of unlabeled data, which is exactly the desired use-case.

This paper presents a new, more efficient entropy gradient derivation and dynamic program that has the same asymptotic time complexity as the gradient for traditional CRF training, $O(ns^2)$. In order to describe this calculation, the paper introduces the concept of *subsequence constrained entropy*—the entropy of a CRF for an observed data sequence when part of the label sequence is fixed. These methods will allow training on larger unannotated data set sizes than previously possible and support active

learning.

2 Semi-Supervised CRF Training

Lafferty et al. (2001) present linear-chain CRFs, a discriminative probabilistic model over observation sequences x and label sequences $Y = \langle Y_1..Y_n \rangle$, where $|x| = |Y| = n$, and each label Y_i has s different possible discrete values. For a linear-chain CRF of Markov order one:

$$p_\theta(Y|x) = \frac{1}{Z(x)} \exp\left(\sum_k \theta_k F_k(x, Y)\right),$$

where $F_k(x, Y) = \sum_i f_k(x, Y_i, Y_{i+1}, i)$, and the partition function $Z(x) = \sum_Y \exp(\sum_k \theta_k F_k(x, Y))$. Given training data $D = \langle d_1..d_n \rangle$, the model is trained by maximizing the log-likelihood of the data $L(\theta; D) = \sum_d \log p_\theta(Y^{(d)}|x^{(d)})$ by gradient methods (e.g. Limited Memory BFGS), where the gradient of the likelihood is:

$$\frac{\partial}{\partial \theta_k} L(\theta; D) = \sum_d F_k(x^{(d)}, Y^{(d)}) - \sum_d \sum_Y p_\theta(Y|x^{(d)}) F_k(x^{(d)}, Y).$$

The second term (the expected counts of the features given the model) can be computed in a tractable amount of time, since according to the Markov assumption, the feature expectations can be rewritten:

$$\sum_Y p_\theta(Y|x) F_k(x, Y) = \sum_i \sum_{Y_i, Y_{i+1}} p_\theta(Y_i, Y_{i+1}|x) f_k(x, Y_i, Y_{i+1}).$$

A dynamic program (the forward/backward algorithm) then computes in time $O(ns^2)$ all the needed probabilities $p_\theta(Y_i, Y_{i+1})$, where n is the sequence length, and s is the number of labels.

For semi-supervised training by *entropy regularization*, we change the objective function by adding the negative entropy of the unannotated data $U = \langle u_1..u_n \rangle$. (Here Gaussian prior is also shown.)

$$L(\theta; D, U) = \sum_n \log p_\theta(Y^{(d)}|x^{(d)}) - \sum_k \frac{\theta_k}{2\sigma^2} + \lambda \sum_u p_\theta(Y^{(u)}|x^{(u)}) \log p_\theta(Y^{(u)}|x^{(u)}).$$

This negative entropy term increases as the decision boundary is moved into sparsely-populated regions of input space.

3 An Efficient Form of the Entropy Gradient

In order to maximize the above objective function, the gradient for the entropy term must be computed. Jiao et al. (2006) perform this computation by:

$$\frac{\partial}{\partial \theta} - H(Y|x) = cov_{p_\theta(Y|x)}[F(x, Y)]\theta,$$

where

$$cov_{p_\theta(Y|x)}[F_j(x, Y), F_k(x, Y)] = E_{p_\theta(Y|x)}[F_j(x, Y), F_k(x, Y)] - E_{p_\theta(Y|x)}[F_j(x, Y)] E_{p_\theta(Y|x)}[F_k(x, Y)].$$

While the second term of the covariance is easy to compute, the first term requires calculation of quadratic feature expectations. The algorithm they propose to compute this term is $O(n^2 s^3)$ as it requires an extra nested loop in forward/backward.

However, the above form of the gradient is not the only possibility. We present here an alternative derivation of the gradient:

$$\begin{aligned}
&\frac{\partial}{\partial \theta_k} - H(Y|x) = \frac{\partial}{\partial \theta_k} \sum_Y p_\theta(Y|x) \log p_\theta(Y|x) \\
&= \sum_Y \left(\frac{\partial}{\partial \theta_k} p_\theta(Y|x)\right) \log p_\theta(Y|x) \\
&+ p_\theta(Y|x) \left(\frac{\partial}{\partial \theta_k} \log p_\theta(Y|x)\right) \\
&= \sum_Y p_\theta(Y|x) \log p_\theta(Y|x) \\
&\qquad \times \left(F_k(x, Y) - \sum_{Y'} p_\theta(Y'|x) F_k(x, Y')\right) \\
&+ \sum_Y p_\theta(Y|x) \left(F_k(x, Y) - \sum_{Y'} p_\theta(Y'|x) F_k(x, Y')\right).
\end{aligned}$$

Since $\sum_Y p_\theta(Y|x) \sum_{Y'} p_\theta(Y'|X) F_k(x, Y') = \sum_{Y'} p_\theta(Y'|X) F_k(x, Y')$, the second summand cancels, leaving:

$$\frac{\partial}{\partial \theta} - H(Y|x) = \sum_Y p_\theta(Y|x) \log p_\theta(Y|x) F_k(x, Y) - \left(\sum_Y p_\theta(Y|x) \log p_\theta(Y|x)\right) \left(\sum_{Y'} p_\theta(Y'|x) F_k(x, Y')\right).$$

Like the gradient obtained by Jiao et al. (2006), there are two terms, and the second is easily computable given the feature expectations obtained by

forward/backward and the entropy for the sequence. However, unlike the previous method, here the first term can be efficiently calculated as well. First, the term must be further factored into a form more amenable to analysis:

$$\begin{aligned}
&\sum_Y p_\theta(Y|x) \log p_\theta(Y|x) F_k(x, Y) \\
&= \sum_Y p_\theta(Y|x) \log p_\theta(Y|x) \sum_i f_k(x, Y_i, Y_{i+1}, i) \\
&= \sum_i \sum_{Y_i, Y_{i+1}} f_k(x, Y_i, Y_{i+1}, i) \\
&\qquad \sum_{Y_{-(i..i+1)}} p_\theta(Y|x) \log p_\theta(Y|x) \,.
\end{aligned}$$

Here, $Y_{-(i..i+1)} = \langle Y_{1..(i-1)} Y_{(i+2)..n} \rangle$. In order to efficiently calculate this term, it is sufficient to calculate $\sum_{Y_{-(i..i+1)}} p_\theta(Y|x) \log p_\theta(Y|x)$ for all pairs y_i, y_{i+1}. The next section presents a dynamic program which can perform these computations in $O(ns^2)$.

4 Subsequence Constrained Entropy

We define *subsequence constrained entropy* as

$$H^\sigma(Y_{-(a..b)}|y_{a..b}, x) = \sum_{Y_{-(a..b)}} p_\theta(Y|x) \log p_\theta(Y|x) \,.$$

The key to the efficient calculation for all subsets is to note that the entropy can be factored given a linear-chain CRF of Markov order 1, since Y_{i+2} is independent of Y_i given Y_{i+1}.

$$\begin{aligned}
&\sum_{Y_{-(a..b)}} p_\theta(Y_{-(a..b)}, y_{a..b}|x) \log p_\theta(Y_{-(a..b)}, y_{a..b}|x) \\
&\quad = \sum_{Y_{-(a..b)}} p_\theta(y_{a..b}|x) p_\theta(Y_{-(a..b)}|y_{a..b}, x) \times \\
&\qquad \big(\log p_\theta(y_{a..b}|x) + \log p_\theta(Y_{-(a..b)}|y_{a..b}, x)\big) \\
&\quad = p_\theta(y_{a..b}|x) \log p_\theta(y_{a..b}|x) \\
&\qquad + p_\theta(y_{a..b}|x) H^\sigma(Y_{-(a..b)}|y_{a..b}, x) \\
&\quad = p_\theta(y_{a..b}|x) \log p_\theta(y_{a..b}|x) \\
&\qquad + p_\theta(y_{a..b}|x) H^\alpha(Y_{1..(a-1)}|y_a, x) \\
&\qquad + p_\theta(y_{a..b}|x) H^\beta(Y_{(b+1)..n}|y_b, x) \,.
\end{aligned}$$

Given the $H^\alpha(\cdot)$ and $H^\beta(\cdot)$ lattices, any sequence entropy can be computed in constant time. Figure 1 illustrates an example in which the constrained sequence is of size two, but the method applies to arbitrary-length contiguous label sequences.

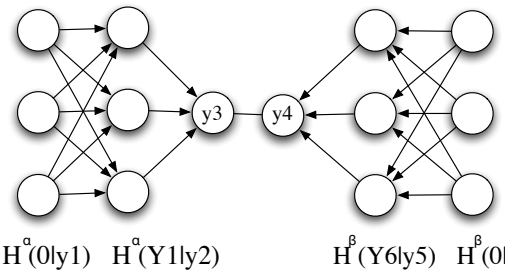

Figure 1: Partial lattice shown for computing the subsequence constrained entropy: $\sum_Y p(Y_{-(3..4)}, y_3, y_4) \log p(Y_{-(3..4)}, y_3, y_4)$. Once the complete H^α and H^β lattices are constructed (in the direction of the arrows), the entropy for each label sequence can be computed in linear time.

Computing the $H^\alpha(\cdot)$ and $H^\beta(\cdot)$ lattices is easily performed using the probabilities obtained by forward/backward. First recall the decomposition formulas for entropy:

$$\begin{aligned}
H(X, Y) &= H(X) + H(Y|X) \\
H(Y|X) &= \sum_x P(X = x) H(Y|X = x) \,.
\end{aligned}$$

Using this decomposition, we can define a dynamic program over the entropy lattices similar to forward/backward:

$$\begin{aligned}
&H^\alpha(Y_{1..i}|y_{i+1}, x) \\
&\quad = H(Y_i|y_{i+1}, x) + H(Y_{1..(i-1)}|Y_i, y_{i+1}, x) \\
&\quad = \sum_{y_i} p_\theta(y_i|y_{i+1}, x) \log p_\theta(y_i|y_{i+1}, x) \\
&\qquad + \sum_{y_i} p_\theta(y_i|y_{i+1}, x) H^\alpha(Y_{1..(i-1)}|y_i) \,.
\end{aligned}$$

The base case for the dynamic program is $H^\alpha(\emptyset|y_1) = p(y_1) \log p(y_1)$. The backward entropy is computed in a similar fashion. The conditional probabilities $p_\theta(y_i|y_{i-1}, x)$ in each of these dynamic programs are available by marginalizing over the per-transition marginal probabilities obtained from forward/backward.

The computational complexity of this calculation for one label sequence requires one run of forward/backward at $O(ns^2)$, and equivalent time to

calculate the lattices for H^α and H^β. To calculate the gradient requires one final iteration over all label pairs at each position, which is again time $O(ns^2)$, but no greater, as forward/backward and the entropy calculations need only to be done once. The complete asymptotic computational cost of calculating the entropy gradient is $O(ns^2)$, which is the same time as supervised training, and a factor of $O(ns)$ faster than the method proposed by Jiao et al. (2006).

Wall clock timing experiments show that this method takes approximately 1.5 times as long as traditional supervised training—less than the constant factors would suggest.[1] In practice, since the three extra dynamic programs do not require recalculation of the dot-product between parameters and input features (typically the most expensive part of inference), they are significantly faster than calculating the original forward/backward lattice.

5 Confidence Estimation

In addition to its merits for computing the entropy gradient, subsequence constrained entropy has other uses, including confidence estimation. Kim et al. (2006) propose using entropy as a confidence estimator in active learning in CRFs, where examples with the most uncertainty are selected for presentation to humans labelers. In practice, they approximate the entropy of the labels given the N-best labels. Not only could our method quickly and exactly compute the true entropy, but it could also be used to find the *sub*sequence that has the highest uncertainty, which could further reduce the additional human tagging effort.

6 Related Work

Hernando et al. (2005) present a dynamic program for calculating the entropy of a HMM, which has some loose similarities to the forward pass of the algorithm proposed in this paper. Notably, our algorithm allows for efficient calculation of entropy for any label subsequence.

Semi-supervised learning has been used in many models, predominantly for classification, as opposed to structured output models like CRFs. Zhu (2005) provides a comprehensive survey of popular semi-supervised learning techniques.

[1] Reporting experimental results with accuracy is unnecessary since we duplicate the training method of Jiao et al. (2006).

7 Conclusion

This paper presents two algorithmic advances. First, it introduces an efficient method for calculating subsequence constrained entropies in linear-chain CRFs, (useful for active learning). Second, it demonstrates how these subsequence constrained entropies can be used to efficiently calculate the gradient of the CRF entropy in time $O(ns^2)$—the same asymptotic time complexity as the forward/backward algorithm, and a $O(ns)$ improvement over previous algorithms—enabling the practical application of CRF *entropy regularization* to large unlabeled data sets.

Acknowledgements

This work was supported in part by DoD contract #HM1582-06-1-2013, in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under NSF grant #IIS-0427594, and in part by the Defense Advanced Research Projects Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under contract number NBCHD030010. Any opinions, findings and conclusions or recommendations expressed in this material belong to the author(s) and do not necessarily reflect those of the sponsor.

References

Y. Grandvalet and Y. Bengio. 2004. Semi-supervised learning by entropy minimization. In *NIPS*.

D. Hernando, V. Crespi, and G. Cybenko. 2005. Efficient computation of the hidden markov model entropy for a given observation sequence. *IEEE Trans. on Information Theory*, 51:7:2681–2685.

F. Jiao, S. Wang, C.-H. Lee, R. Greiner, and D. Schuurmans. 2006. Semi-supervised conditional random fields for improved sequence segmentation and labeling. In *COLING/ACL*.

S. Kim, Y. Song, K. Kim, J.-W. Cha, and G. G. Lee. 2006. Mmr-based active machine learning for bio named entity recognition. In *HLT/NAACL*.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*, pages 282–289.

X. Zhu. 2005. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison.