Softmax-Margin CRFs: Training Log-Linear Models with Cost Functions

Kevin Gimpel Noah A. Smith
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{kgimpel,nasmith}@cs.cmu.edu

Abstract

We describe a method of incorporating task-specific cost functions into standard conditional log-likelihood (CLL) training of linear structured prediction models. Recently introduced in the speech recognition community, we describe the method generally for structured models, highlight connections to CLL and max-margin learning for structured prediction (Taskar et al., 2003), and show that the method optimizes a bound on risk. The approach is simple, efficient, and easy to implement, requiring very little change to an existing CLL implementation. We present experimental results comparing with several commonly-used methods for training structured predictors for named-entity recognition.

1 Introduction

Conditional random fields (CRFs; Lafferty et al, 2001) and other conditional log-linear models (Berger et al., 1996) achieve strong performance for many NLP problems, but the conditional log-likelihood (CLL) criterion optimized when training these models cannot take a task-specific cost function into account.

In this paper, we describe a simple approach for training conditional log-linear models with cost functions. We show how the method relates to other methods and how it provides a bound on risk. We apply the method to train a discriminative model for named-entity recognition, showing a statistically significant improvement over CLL.

2 Structured Log-Linear Models

Let $\mathcal{X}$ denote a structured input space and, for a particular $x \in \mathcal{X}$, let $\mathcal{Y}(x)$ denote a structured output space for x. The size of $\mathcal{Y}(x)$ is often exponential in x, which differentiates structured prediction from multiclass classification. For named-entity recognition, for example, x might be a sentence and $\mathcal{Y}(x)$ the set of all possible named-entity labelings for the sentence. Given an $x \in \mathcal{X}$ and a $y \in \mathcal{Y}(x)$, we use a conditional log-linear model for $p_{\boldsymbol{\theta}}(y|x)$:

$$p_{\boldsymbol{\theta}}(y|x) = \frac{\exp\{\boldsymbol{\theta}^\top \boldsymbol{f}(x,y)\}}{\sum_{y' \in \mathcal{Y}(x)} \exp\{\boldsymbol{\theta}^\top \boldsymbol{f}(x,y')\}} \quad (1)$$

where $\boldsymbol{f}(x,y)$ is a feature vector representation of x and y and $\boldsymbol{\theta}$ is a parameter vector containing one component for each feature.

2.1 Training Criteria

Many criteria exist for training the weights $\boldsymbol{\theta}$. We next review three choices in detail. For the following, we assume a training set consisting of n examples $\{\langle x^{(i)}, y^{(i)}\rangle\}_{i=1}^n$. Some criteria will make use of a task-specific cost function that measures the extent to which a structure y differs from the true structure $y^{(i)}$, denoted by $\mathrm{cost}(y^{(i)}, y)$.

2.1.1 Conditional Log-Likelihood

The learning problem for maximizing conditional log-likelihood is shown in Eq. 3 in Fig. 1 (we transform it into a minimization problem for easier comparison). This criterion is commonly used when a probabilistic interpretation of the model is desired.

2.1.2 Max-Margin

An alternative approach to training structured linear classifiers is based on maximum-margin Markov networks (Taskar et al., 2003). The basic idea is to choose weights such that the linear score of each $\langle x^{(i)}, y^{(i)}\rangle$ is better than $\langle x^{(i)}, y\rangle$ for all alternatives $y \in \mathcal{Y}(x^{(i)}) \setminus \{y^{(i)}\}$, with a larger margin for those y with higher cost. The "margin rescaling" form of this training criterion is shown in Eq. 4. Note that the cost function is incorporated into the criterion.

2.1.3 Risk

Risk is defined as the expected value of the cost with respect to the conditional distribution $p_{\boldsymbol{\theta}}(y|x)$;

on training data:

$$\sum_{i=1}^{n} \sum_{y \in \mathcal{Y}(x^{(i)})} p_{\boldsymbol{\theta}}(y|x^{(i)}) \text{cost}(y^{(i)}, y) \quad (2)$$

With a log-linear model, learning then requires solving the problem shown in Eq. 5. Unlike the previous two criteria, risk is typically non-convex.

Risk minimization first appeared in the speech recognition community (Kaiser et al., 2000; Povey and Woodland, 2002). In NLP, Smith and Eisner (2006) minimized risk using k-best lists to define the distribution over output structures. Li and Eisner (2009) introduced a novel semiring for minimizing risk using dynamic programming; Xiong et al. (2009) minimized risk in a CRF.

2.1.4 Other Criteria

Many other criteria have been proposed to attempt to tailor training conditions to match task-specific evaluation metrics. These include the average per-label marginal likelihood for sequence labeling (Kakade et al., 2002), minimum error-rate training for machine translation (Och, 2003), F_1 for logistic regression classifiers (Jansche, 2005), and a wide range of possible metrics for sequence labeling and segmentation tasks (Suzuki et al., 2006).

3 Softmax-Margin

The softmax-margin objective is shown as Eq. 6 and is a generalization of that used by Povey et al. (2008) and similar to that used by Sha and Saul (2006). The simple intuition is the same as the intuition in max-margin learning: high-cost outputs for $x^{(i)}$ should be penalized more heavily. Another view says that we replace the probabilistic score inside the $\exp$ function of CLL with the "cost-augmented" score from max-margin. A third view says that we replace the "hard" maximum of max-margin with the "softmax" ($\log \sum \exp$) from CLL; hence we use the name "softmax-margin." Like CLL and max-margin, the objective is convex; a proof is provided in Gimpel and Smith (2010).

3.1 Relation to Other Objectives

We next show how the softmax-margin criterion (Eq. 6) bounds the risk criterion (Eq. 5). We first define some additional notation:

$$\mathbb{E}_{(i)}[F] \quad = \quad \sum_{y \in \mathcal{Y}(x^{(i)})} p_{\boldsymbol{\theta}}(y \mid x^{(i)}) F(y)$$

for some function $F : \mathcal{Y}(x^{(i)}) \rightarrow \mathbb{R}$. First note that the softmax-margin objective (Eq. 6) is equal to:

$$\text{(Eq. 3)} + \sum_{i=1}^{n} \log \mathbb{E}_{(i)}[\exp \text{cost}(y^{(i)}, \cdot)] \quad (7)$$

The first term must be nonnegative. Taking each part of the second term, and using Jensen's inequality,

$$\begin{aligned} \log \mathbb{E}_{(i)}[e^{\text{cost}(y^{(i)}, \cdot)}] \quad &\geq \quad \mathbb{E}_{(i)}[\log e^{\text{cost}(y^{(i)}, \cdot)}] \\ &= \quad \mathbb{E}_{(i)}[\text{cost}(y^{(i)}, \cdot)] \end{aligned}$$

which is exactly Eq. 5. Softmax-margin is also an upper bound on the CLL criterion because, assuming cost is nonnegative, $\log \mathbb{E}[\exp \text{cost}] \geq 0$. Furthermore, softmax-margin is a differentiable upper bound on max-margin, because the softmax function is a differentiable upper bound on the max function.

We note that it may also be interesting to consider minimizing the function $\sum_{i=1}^{n} \log \mathbb{E}_{(i)}[\exp \text{cost}(y^{(i)}, \cdot)]$, since it is an upper bound on risk but requires less computation for computing the gradient.[1] We call this objective the **Jensen risk bound** and include it in our experimental comparison below.

3.2 Implementation

Most methods for training structured models with cost functions require the cost function to decompose across the pieces of the structure in the same way as the features, such as the standard methods for maximizing margin and minimizing risk (Taskar et al., 2003; Li and Eisner, 2009). If the same conditions hold, softmax-margin training can be implemented atop standard CRF training simply by adding additional "features" to encode the local cost components, *only* when computing the partition function during training.[2] The weights of these "cost features" are not learned.

4 Experiments

We consider the problem of named-entity recognition (NER) and use the English data from the CoNLL 2003 shared task (Tjong Kim Sang and De Meulder, 2003). The data consist of news articles

[1] Space does not permit a full discussion; see Gimpel and Smith (2010) for details.

[2] Since $\text{cost}(y^{(i)}, y^{(i)}) = 0$ by definition, these "features" will never fire for the numerator and can be ignored.

$$\text{CLL:} \quad \min_{\boldsymbol{\theta}} \quad \sum_{i=1}^{n} -\boldsymbol{\theta}^\top \boldsymbol{f}(x^{(i)}, y^{(i)}) + \log \sum_{y \in \mathcal{Y}(x^{(i)})} \exp\{\boldsymbol{\theta}^\top \boldsymbol{f}(x^{(i)}, y)\} \tag{3}$$

$$\text{Max-Margin:} \quad \min_{\boldsymbol{\theta}} \quad \sum_{i=1}^{n} -\boldsymbol{\theta}^\top \boldsymbol{f}(x^{(i)}, y^{(i)}) + \max_{y \in \mathcal{Y}(x^{(i)})} \left(\boldsymbol{\theta}^\top \boldsymbol{f}(x^{(i)}, y) + \text{cost}(y^{(i)}, y)\right) \tag{4}$$

$$\text{Risk:} \quad \min_{\boldsymbol{\theta}} \quad \sum_{i=1}^{n} \sum_{y \in \mathcal{Y}(x^{(i)})} \text{cost}(y^{(i)}, y) \frac{\exp\{\boldsymbol{\theta}^\top \boldsymbol{f}(x^{(i)}, y)\}}{\sum_{y' \in \mathcal{Y}(x^{(i)})} \exp\{\boldsymbol{\theta}^\top \boldsymbol{f}(x^{(i)}, y')\}} \tag{5}$$

$$\text{Softmax-Margin:} \quad \min_{\boldsymbol{\theta}} \quad \sum_{i=1}^{n} -\boldsymbol{\theta}^\top \boldsymbol{f}(x^{(i)}, y^{(i)}) + \log \sum_{y \in \mathcal{Y}(x^{(i)})} \exp\{\boldsymbol{\theta}^\top \boldsymbol{f}(x^{(i)}, y) + \text{cost}(y^{(i)}, y)\} \tag{6}$$

Figure 1: Objective functions for training linear models. Regularization terms (e.g., $C \sum_{j=1}^{d} \theta_j^2$) are not shown here.

annotated with four entity types: person, location, organization, and miscellaneous. Our experiments focus on comparing training objectives for structured sequential models for this task. For all objectives, we use the same standard set of feature templates, following Kazama and Torisawa (2007) with additional token shape like those in Collins (2002b) and simple gazetteer features. A feature was included if it occurred at least once in training data (total 1,312,255 features).

The task is evaluated using the F_1 score, which is the harmonic mean of precision and recall (computed at the level of entire entities). Since this metric is computed from corpus-level precision and recall, it is not easily decomposable into features used in standard chain CRFs. For simplicity, we only consider Hamming cost in this paper; experiments with other cost functions more targeted to NER are presented in Gimpel and Smith (2010).

4.1 Baselines

We compared softmax-margin to several baselines: the structured perceptron (Collins, 2002a), 1-best MIRA with cost-augmented inference (Crammer et al., 2006), CLL, max-margin, risk, and our Jensen risk bound (JRB) introduced above.

We used L_2 regularization, experimenting with several coefficients for each method. For CLL, softmax-margin, max-margin, and MIRA, we used regularization coefficients $C \in \{0.01, 0.1, 1\}$. Risk has not always been used with regularization, as regularization does not have as clear a probabilistic interpretation with risk as it does with CLL; so, for risk and JRB we only used $C \in \{0.0, 0.01\}$. In addition, since these two objectives are non-convex, we initialized with the output of the best-performing CLL model on dev data (which was the CLL model with $C = 0.01$).[3] All methods except CLL and the perceptron make use of a cost function, for which we used Hamming cost. We experimented with different fixed multipliers m for the cost function, for $m \in \{1, 5, 10, 20\}$.

The hyperparameters C and m were tuned on the development data and the best-performing combination was used to label the test data. We also tuned the decision to average parameters across all training iterations; this has generally been found to help the perceptron and MIRA, but in our experiments had mixed results for the other methods.

We ran 100 iterations through the training data for each method. For CLL, softmax-margin, risk, and JRB, we used stochastic gradient ascent with a fixed step size of 0.01. For max-margin, we used stochastic subgradient ascent (Ratliff et al., 2006) also with a fixed step size of 0.01.[4] For the perceptron and MIRA, we used their built-in step size formulas.

4.2 Results

Table 1 shows our results. On test data, softmax-margin is statistically indistinguishable from MIRA, risk, and JRB, but performs significantly better than CLL, max-margin, and the perceptron ($p < 0.03$, paired bootstrap with 10,000 samples; Koehn,

[3] When using initialization of all ones for risk and JRB, results were several points below the results here, and with all zeroes, learning failed, resulting in 0.0 F-measure on dev data. Thus, risk and JRB appear sensitive to model initialization.

[4] In preliminary experiments, we tried other fixed and decreasing step sizes for (sub)gradient ascent and found that a fixed step of 0.01 consistently performed well across training objectives, so we used it for all settings for simplicity.

Method	Dev.	Test	(C, m, avg.?)
Perceptron	90.48	83.98	(Y)
MIRA	91.13	85.72	(0.01, 20, Y)
CLL	90.79	85.46	(0.01, N)
Max-Margin	91.17	85.28	(0.01, 1, Y)
Risk	91.14	85.59	(0.01, 10, N)
JRB	91.05	85.65	(0.01, 1, N)
Softmax-Margin	91.30	85.84	(0.01, 5, N)

Table 1: Results on development and test sets, along with hyperparameter values chosen using development set.

2004). It may be surprising that an improvement of 0.38 in F_1 could be significant, but this indicates that the improvements are not limited to certain categories of phenomena in a small number of sentences but rather appear throughout the majority of the test set. The Jensen risk bound performs comparably to risk, and takes roughly half as long to train.

5 Discussion

The softmax-margin approach offers (1) a convex objective, (2) the ability to incorporate task-specific cost functions, and (3) a probabilistic interpretation (which supports, e.g., hidden-variable learning and computation of posteriors). In contrast, max-margin training and MIRA do not provide (3); risk and JRB do not provide (1); and CLL does not support (2). Furthermore, softmax-margin training improves over standard CLL training of CRFs, is straightforward to implement, and requires the same amount of computation as CLL.

We have also presented the Jensen risk bound, which is easier to implement and faster to train than risk, yet gives comparable performance. The primary limitation of all these approaches, including softmax-margin, is that they only support cost functions that factor in the same way as the features of the model. Future work might exploit approximate inference for more expressive cost functions.

Acknowledgments

We thank the reviewers, John Lafferty, and André Martins for helpful comments and feedback on this work. This research was supported by NSF grant IIS-0844507.

References

A. Berger, V. J. Della Pietra, and S. A. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

M. Collins. 2002a. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proc. of EMNLP*.

M. Collins. 2002b. Ranking algorithms for named-entity extraction: Boosting and the voted perceptron. In *Proc. of ACL*.

K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585.

K. Gimpel and N. A. Smith. 2010. Softmax-margin training for structured log-linear models. Technical report, Carnegie Mellon University.

M. Jansche. 2005. Maximum expected F-measure training of logistic regression models. In *Proc. of HLT-EMNLP*.

J. Kaiser, B. Horvat, and Z. Kacic. 2000. A novel loss function for the overall risk criterion based discriminative training of HMM models. In *Proc. of ICSLP*.

S. Kakade, Y. W. Teh, and S. Roweis. 2002. An alternate objective function for Markovian fields. In *Proc. of ICML*.

J. Kazama and K. Torisawa. 2007. A new perceptron algorithm for sequence labeling with non-local features. In *Proc. of EMNLP-CoNLL*.

P. Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proc. of EMNLP*.

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

Z. Li and J. Eisner. 2009. First- and second-order expectation semirings with applications to minimum-risk training on translation forests. In *Proc. of EMNLP*.

F. J. Och. 2003. Minimum error rate training for statistical machine translation. In *Proc. of ACL*.

D. Povey and P. C. Woodland. 2002. Minimum phone error and I-smoothing for improved discriminative training. In *Proc. of ICASSP*.

D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah. 2008. Boosted MMI for model and feature space discriminative training. In *Proc. of ICASSP*.

N. Ratliff, J. A. Bagnell, and M. Zinkevich. 2006. Subgradient methods for maximum margin structured learning. In *ICML Workshop on Learning in Structured Output Spaces*.

F. Sha and L. K. Saul. 2006. Large margin hidden Markov models for automatic speech recognition. In *Proc. of NIPS*.

D. A. Smith and J. Eisner. 2006. Minimum risk annealing for training log-linear models. In *Proc. of COLING-ACL*.

J. Suzuki, E. McDermott, and H. Isozaki. 2006. Training conditional random fields with multivariate evaluation measures. In *Proc. of COLING-ACL*.

B. Taskar, C. Guestrin, and D. Koller. 2003. Max-margin Markov networks. In *Advances in NIPS 16*.

E. F. Tjong Kim Sang and F. De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proc. of CoNLL*.

Y. Xiong, J. Zhu, H. Huang, and H. Xu. 2009. Minimum tag error for discriminative training of conditional random fields. *Information Sciences*, 179(1-2):169–179.