

Constraint-Driven Rank-Based Learning for Information Extraction

Sameer Singh Limin Yao Sebastian Riedel Andrew McCallum

Dept. of Computer Science
University of Massachusetts
Amherst MA 01003

{sameer, lmyao, riedel, mccallum}@cs.umass.edu

Abstract

Most learning algorithms for undirected graphical models require complete inference over at least one instance before parameter updates can be made. SampleRank is a rank-based learning framework that alleviates this problem by updating the parameters during inference. Most semi-supervised learning algorithms also perform full inference on at least one instance before each parameter update. We extend SampleRank to semi-supervised learning in order to circumvent this computational bottleneck. Different approaches to incorporate unlabeled data and prior knowledge into this framework are explored. When evaluated on a standard information extraction dataset, our method significantly outperforms the supervised method, and matches results of a competing state-of-the-art semi-supervised learning approach.

1 Introduction

Most supervised learning algorithms for undirected graphical models require full inference over the dataset (e.g., gradient descent), small subsets of the dataset (e.g., stochastic gradient descent), or at least a single instance (e.g., perceptron, Collins (2002)) before parameter updates are made. Often this is the main computational bottleneck during training.

SampleRank (Wick et al., 2009) is a rank-based learning framework that alleviates this problem by performing parameter updates *within* inference. Every pair of samples generated during inference is ranked according to the model and the ground truth, and the parameters are updated when the rankings disagree. SampleRank has enabled efficient learn-

ing for massive information extraction tasks (Culotta et al., 2007; Singh et al., 2009).

The problem of requiring a complete inference iteration before parameters are updated also exists in the semi-supervised learning scenario. Here the situation is often considerably worse since inference has to be applied to potentially very large unlabeled datasets. Most semi-supervised learning algorithms rely on marginals (GE, Mann and McCallum, 2008) or MAP assignments (CODL, Chang et al., 2007). Calculating these is computationally inexpensive for many simple tasks (such as classification and regression). However, marginal and MAP inference tends to be expensive for complex structured prediction models (such as the joint information extraction models of Singh et al. (2009)), making semi-supervised learning intractable.

In this work we employ a fast rank-based learning algorithm for semi-supervised learning to circumvent the inference bottleneck. The ranking function is extended to capture both the preference expressed by the labeled data, and the preference of the domain expert when the labels are not available. This allows us to perform SampleRank as is, without sacrificing its scalability, which is crucial for future large scale applications of semi-supervised learning.

We applied our method to a standard information extraction dataset used for semi-supervised learning. Empirically we demonstrate improvements over the supervised model, and closely match the results of a competing state-of-the-art semi-supervised learner.

2 Background

Conditional random fields (Lafferty et al., 2001) are undirected graphical models represented as factor

graphs. A factor graph $G = \{\Psi_i\}$ defines a probability distribution over assignments \mathbf{y} to a set of output variables, conditioned on an observation \mathbf{x} . A factor Ψ_i computes the inner product between the vector of sufficient statistics $\mathbf{f}(\mathbf{x}_i, \mathbf{y}_i)$ and parameters Θ . Let $Z(\mathbf{x})$ be the data-dependent partition function used for normalization. The probability distribution defined by the graph is:

$$p(\mathbf{y}|\mathbf{x}, \Theta) = \frac{1}{Z(\mathbf{x})} \prod_{\Psi_i \in G} e^{\Theta \cdot \mathbf{f}(\mathbf{x}_i, \mathbf{y}_i)}$$

2.1 Rank-Based Learning

SampleRank (Wick et al., 2009) is a rank-based learning framework for that performs parameter updates *within* MCMC inference. Every pair of consecutive samples in the MCMC chain is ranked according to the model and the ground truth, and the parameters are updated when the rankings disagree. This allows the learner to acquire more supervision per sample, and has led to efficient training of models for which inference is very expensive (Singh et al., 2009).

SampleRank considers two ranking functions: (1) the unnormalized conditional probability (model ranking), and (2) a *truth function* $\mathcal{F}(\mathbf{y})$ (objective ranking) which is defined as $-\mathcal{L}(\mathbf{y}, \mathbf{y}_L)$, the negative loss between the possible assignment \mathbf{y} and the true assignment \mathbf{y}_L . The truth function can take different forms, such as tokenwise accuracy or F1-measure with respect to some labeled data.

In order to learn the parameters for which model rankings are consistent with objective rankings, SampleRank performs the following update for each consecutive pair of samples \mathbf{y}^a and \mathbf{y}^b of the MCMC chain. Let α be the learning rate, and $\Delta = \mathbf{f}(\mathbf{x}_i, \mathbf{y}_i^a) - \mathbf{f}(\mathbf{x}_i, \mathbf{y}_i^b)$, then Θ is updated as follows:

$$\Theta \leftarrow \begin{cases} \alpha\Delta & \text{if } \frac{p(\mathbf{y}^a|\mathbf{x})}{p(\mathbf{y}^b|\mathbf{x})} < 1 \wedge \mathcal{F}(\mathbf{y}^a) > \mathcal{F}(\mathbf{y}^b) \\ -\alpha\Delta & \text{if } \frac{p(\mathbf{y}^a|\mathbf{x})}{p(\mathbf{y}^b|\mathbf{x})} > 1 \wedge \mathcal{F}(\mathbf{y}^a) < \mathcal{F}(\mathbf{y}^b) \\ 0 & \text{otherwise.} \end{cases}$$

This update is usually fast: in order to calculate the required model ratio, only factors that touch changed variables have to be taken into account.

SampleRank has been incorporated into the FACTORIE toolkit for probabilistic programming with imperatively-defined factor graphs (McCallum et al., 2009).

3 Semi-Supervised Rank-Based Learning

To apply SampleRank to the semi-supervised setting, we need to specify the truth function \mathcal{F} over both labeled and unlabeled data. For labeled data \mathcal{Y}_L , we can use the true labels. These are not available for unlabeled data \mathcal{Y}_U , and we present alternative ways of defining a truth function $\mathcal{F}_U : \mathcal{Y}_U \rightarrow \mathfrak{R}$ for this case.

3.1 Self-Training

Self-training, which uses predictions as truth, fits directly into our SampleRank framework. After performing SampleRank on training data (using \mathcal{F}_L), MAP inference is performed on the unlabeled data. The prediction $\hat{\mathbf{y}}_U$ is used as the ground truth for the unlabeled data. Thus the self-training objective function \mathcal{F}_s over the unlabeled data can be defined as $\mathcal{F}_s(\mathbf{y}) = -\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}_U)$.

3.2 Encoding Constraints

Constraint-driven semi-supervised learning uses constraints to incorporate external domain knowledge when labels are missing (Chang et al., 2007; Mann and McCallum, 2008; Bellare et al., 2009). Constraints prefer certain label configurations over others. For example, one constraint may be that occurrences of the word ‘‘California’’ are preferred to have the label ‘‘location’’.

We can encode constraints directly into the objective function \mathcal{F}_U . Let a constraint i be specified as $\langle p_i, c_i \rangle$, where $c_i(\mathbf{y})$ denotes whether assignment \mathbf{y} satisfies the constraint i (+1), violates it (−1), or the constraint does not apply (0), and p_i is the constraint strength. Then the objective function is:

$$\mathcal{F}_c(\mathbf{y}) = \sum_i p_i c_i(\mathbf{y})$$

3.3 Incorporating Model Predictions

When the objective function \mathcal{F}_c is used, every prediction on unlabeled data is ranked only according to the constraints, and thus the model is trained to satisfy all the constraints. This is a problem when the constraints prefer a wrong solution while the model favors the correct solution, resulting in SampleRank updating the model away from the true solution. To avoid this, the ranking function needs to balance preferences of the constraints and the current model.

One option is to incorporate the self-training objective function \mathcal{F}_s . A new objective function that combines self-training with constraints can be defined as:

$$\begin{aligned}\mathcal{F}_{sc}(\mathbf{y}) &= \mathcal{F}_s(\mathbf{y}) + \lambda_s \mathcal{F}_c(\mathbf{y}) \\ &= -\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}_U) + \lambda_s \sum_i p_i c_i(\mathbf{y})\end{aligned}$$

This objective function has at least two limitations. First, self-training involves a complete inference step to obtain $\hat{\mathbf{y}}_U$. Second, the model might have low confidence in its prediction (this is the case when the underlying marginals are almost uniform), but the self-training objective does not take this into account. Hence, we also propose an objective function that incorporates the model score directly, i.e.

$$\begin{aligned}\mathcal{F}_{mc}(\mathbf{y}) &= \log p(\mathbf{y}|\mathbf{x}, \Theta) + \log Z(x) + \lambda_m \mathcal{F}_c(\mathbf{y}) \\ &= \sum_{\Psi_i} \Theta \cdot \mathbf{f}(\mathbf{x}_i, \mathbf{y}_i) + \lambda_m \sum_i p_i c_i(\mathbf{y})\end{aligned}$$

This objective does not require inference, and also takes into account model confidence.

In both objective functions \mathcal{F}_{sc} and \mathcal{F}_{mc} , λ controls the relative contribution of the constraint preferences to the objective function. With higher λ , SampleRank will make updates that never try to violate constraints, while with low λ , SampleRank trusts the model more. λ corresponds to constraint satisfaction weights ρ used in (Chang et al., 2007).

4 Related Work

Chang et al. propose constraint-driven learning (CODL, Chang et al., 2007) which can be interpreted as a variation of self-training: Instances are selected for supervision based not only on the model’s prediction, but also on their consistency with a set of user-defined constraints. By directly incorporating the model score and the constraints (as in \mathcal{F}_{mc} in Section 3.3) we follow the same approach, but avoid the expensive “Top-K” inference step.

Generalized expectation criterion (GE, Mann and McCallum, 2008) and Alternating Projections (AP, Bellare et al., 2009) encode preferences by specifying constraints on feature expectations, which require expensive inference. Although AP can use online training, it still involves full inference over each

instance. Furthermore, these methods only support constraints that factorize according to the model.

Li (2009) incorporates prior knowledge into conditional random fields as variables. They require full inference during learning, restricting the application to simple models. Furthermore, higher-order constraints are specified using large cliques in the graph, which slow down inference. Our approach directly incorporates these constraints into the ranking function, with no impact on inference time.

5 Experiments

We carried out experiments on the Cora citation dataset. The task is to segment each citation into different fields, such as “author” and “title”. We use 300 instances as training data, 100 instances as development data, and 100 instances as test data. Some instances from the training data are selected as labeled instances, and the remaining data (including development) as unlabeled. We use the same token-label constraints as Chang et al. (2007).

We use the objective functions defined in Section 3, specifically self-training (Self: \mathcal{F}_s), direct constraints (Cons: \mathcal{F}_c), the combination of the two (Self+Cons: \mathcal{F}_{sc}), and combination of the model score and the constraints (Model+Cons: \mathcal{F}_{mc}). We set $p_i = 1.0$, $\alpha = 1.0$, $\lambda_s = 10$, and $\lambda_m = 0.0001$.

Average token accuracy for 5 runs is reported and compared with CODL¹ in Table 1. We also report supervised results from (Chang et al., 2007) and SampleRank. All of our methods show vast improvement over the supervised method for smaller training sizes, but this difference decreases as the training size increases. When the complete training data is used, additional unlabeled data hurts our performance. This is not observed in CODL since they use more unlabeled data, which may also explain their slightly higher accuracy. Note that Self+Cons performs better than Self or Cons individually.

Model+Cons also performs competitively, and may potentially outperform other methods if a better λ_m is chosen. Note, however, that λ_m is much harder to tune than λ_s since λ_m weighs the contribution of the unnormalized model score, the range

¹We report *inference without constraints* results from CODL. Their results that incorporated constraints were higher, but we do not implement this alternative due to the difficulty in balancing the model score and constraint weights.

Method	5	10	15	20	25	300
Sup. (CODL)	55.1	64.6	68.7	70.1	72.7	86.1
SampleRank	66.5	74.6	75.6	77.6	79.5	90.7
CODL	71	76.7	79.4	79.4	82	88.2
Self	67.6	75.1	75.8	78.6	80.4	88
Cons	67.2	75.3	77.5	78.6	79.4	88.3
Self+Cons	71.3	77	77.5	79.5	81.1	87.4
Model+Cons	69.8	75.4	75.7	79.3	79.3	90.6

Table 1: **Tokenwise Accuracy:** for different methods as we vary the size of the labeled data

of which depends on many different factors such as properties of the data, the learning rate, number of samples, proposal function, etc. For self+cons (λ_s), the ranges of the predictions and constraint penalties are fixed and known, making the task simpler.

Self training takes 90 minutes to run on average, while Self+Cons and Model+Cons need 100 minutes. Since the Cons method skips the inference step over unlabeled data, it takes only 30 minutes to run. As the size of the model and unlabeled data set grows, this saving will become more significant. Running time of CODL was not reported.

6 Conclusion

This work extends the rank-based learning framework to semi-supervised learning. By integrating the two paradigms, we retain the computational efficiency provided by parameter updates *within inference*, while utilizing unlabeled data and prior knowledge. We demonstrate accuracy improvements on a real-word information extraction dataset.

We believe that the method will be of greater benefit to learning in complex factor graphs such as joint models over multiple extraction tasks. In future work we will investigate our approach in such settings. Additionally, various sensitivity, convergence, and robustness properties of the method need to be analyzed.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval, in part by SRI International subcontract #27-001338 and ARFL prime contract #FA8750-09-C-0181, and in part by The Central Intelligence Agency, the National Security Agency and National Science Foundation under

NSF grant #IIS-0326249. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

References

- Kedar Bellare, Gregory Druck, and Andrew McCallum. Alternating projections for learning with expectation constraints. In *UAI*, 2009.
- Mingwei Chang, Lev Ratinov, and Dan Roth. Guiding semi-supervision with constraint-driven learning. In *ACL*, 2007.
- Michael Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithm. In *ACL*, 2002.
- Aron Culotta, Michael Wick, and Andrew McCallum. First-order probabilistic models for coreference resolution. In *NAACL/HLT*, 2007.
- John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.
- Xiao Li. On the use of virtual evidence in conditional random fields. In *EMNLP*, 2009.
- Gideon S. Mann and Andrew McCallum. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *ACL*, 2008.
- Andrew McCallum, Karl Schultz, and Sameer Singh. FACTORIE: probabilistic programming via imperatively defined factor graphs. In *NIPS*, 2009.
- Sameer Singh, Karl Schultz, and Andrew McCallum. Bi-directional joint inference for entity resolution and segmentation using imperatively-defined factor graphs. In *ECML/PKDD*, 2009.
- Michael Wick, Khashayar Rohanimanesh, Aron Culotta, and Andrew McCallum. SampleRank: Learning preferences from atomic gradients. In *NIPS Workshop on Advances in Ranking*, 2009.