2010

Constraint-Driven Rank-Based Learning for Information Extraction

Sameer Singh
Limin Yao
Sebastian Riedel
Andrew McCallum, *University of Massachusetts - Amherst*

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 factor graphs require complete inference over the dataset or an instance before making an update to the parameters. SampleRank is a rank-based learning framework that alleviates this problem by updating the parameters during inference. Most semi-supervised learning algorithms also rely on the complete inference, i.e. calculating expectations or MAP configurations. We extend the SampleRank framework to the semi-supervised learning, avoiding these inference bottlenecks. Different approaches for incorporating unlabeled data and prior knowledge into this framework are explored. We evaluated our method on a standard information extraction dataset. Our approach outperforms the supervised method significantly and matches the result of the competing semi-supervised learning approach.

1 Introduction

Most supervised learning algorithms for factor graphs require full inference over the dataset (*e.g.* conditional loglikelihood) or an instance (*e.g.* Collin's perceptron) before parameter updates can be made. Often this is the main computational bottleneck during training.

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

The problem of requiring a complete inference iteration before the parameters are updates also exists in the semi-supervised learning scenario, where the situation is worse compared to supervised learning, as the inference has to be applied to the large unlabeled dataset. Most semi-supervised learning algorithms are designed to address many machine learning tasks and rely on marginals (GE: (Mann and McCallum, 2008)) or MAP assignments (CODL: (Chang et al., 2007)) which are extremely cheap for many of these tasks (such as classification and regression). However, marginal and MAP inference is often intractable for the factor graphs used for information extraction.

This work employs the fast rank-based learning algorithm for semi-supervised learning on factor graphs to avoid the inference bottleneck. We demonstrate how SampleRank naturally extends to semi-supervised learning by generalizing the notion of rank constraints to include both preference expressed as the labeled data, and preference of the model designer when the labels are not available. This allows us to perform SampleRank as is, without sacrificing its scalability, which is necessary 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 more complex competing semi-supervised learner,

while running significantly faster.

2 Background

A factor graph G defines a probability distribution over assignments $\mathbf{y}$ to a set of output variables, conditioned on input variables $\mathbf{x}$. 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 conditional probability distribution defined by the factor graph is:

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

2.1 Rank-Based Learning

Most learning methods need to calculate the model expectations (Lafferty et al., 2001) or the MAP configuration (Collins, 2002) before making an update to the parameters. This step of *inference* is usually the bottleneck for learning, even when performed approximately.

SampleRank (Rohanimanesh et al., 2009) is a rank-based learning framework that alleviates this problem by performing parameter updates *within* MCMC inference. Every pair of consectutive 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 instance, and has led to efficient training for models in which inference are expensive and generally intractable (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 $\mathbf{y}_L$. One such truth function is tokenwise accuracy with respect to some labeled data, another could be the F1-measure.

In order to learn parameters for which model rankings are consistent with objective rankings, SampleRank performs the following update at each step of the MCMC chain. Given two samples $\mathbf{y}^a$ and $\mathbf{y}^b$, let α be the learning rate, and $\Delta = \mathbf{f}(\mathbf{x}_i, \mathbf{y}_i^a) - \mathbf{f}(\mathbf{x}_i, \mathbf{y}_i^b)$, the weights are changed as follows:

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

Calculating these rankings does not require inference.

Note that it has recently been incorporated as part of the imperatively-defined factor graphs (IDFs) in the FACTORIE toolkit (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, however these are not available for unlabeled data $\mathcal{Y}_U$. Inspired by semi-supervised learning framework, there are several different ways of defining the truth function $\mathcal{F}_U : \mathcal{Y}_U \rightarrow \Re$ over unlabeled data.

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

Recent research on constraint-driven semi-supervised learning uses various constraints to specify external domain knowledge (Chang et al., 2007; Mann and McCallum, 2008; Bellare et al., 2009). These methods thus integrate labeled data, unlabeled data, and constraints into one unified framework.

For example, a constraint on a token may capture its preference for a particular label, i.e. token "NY" prefers being labeled as "location" with high confidence. If a labeling satisfies this constraint, the constraint-based objective function should score it higher than a labeling that violates this constraint.

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 strength associated with the constraint. Then,

$$\mathcal{F}_c(\mathbf{y}) = \sum_i p_i c_i(\mathbf{y}) \quad (2)$$

3.3 Incorporating Model Predictions

When objective function $\mathcal{F}_c$ is used, every change to unlabeled data is ranked only according to the constraints, and thus the model will attempt to satisfy all the constraints. To allow soft constraints, the model's current state has to be taken into account.

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

$$\begin{aligned} \mathcal{F}_{sc}(\mathbf{y}) &= \mathcal{F}_s(\mathbf{y}) + \lambda_s \mathcal{F}_c(\mathbf{y}) \quad (3) \\ &= -\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}_U) + \lambda_s \sum_i p_i c_i(\mathbf{y}) \end{aligned}$$

This objective function has several limitations. First, self-training involves a complete inference step to obtain $\hat{\mathbf{y}}_U$. Second, the predictions are from an older model, which may be obsolete. Instead, we propose another objective function that incorporates the model score directly into the objective function, i.e.

$$\begin{aligned} \mathcal{F}_{mc}(\mathbf{y}) &= \log p(\mathbf{y}|\mathbf{x}, \Theta) + \log Z(x) + \lambda_m \mathcal{F}_c(\mathbf{y}) \\ &= \sum_{\Psi_i} \Theta \cdot \mathbf{f}(\mathbf{x}_i, \mathbf{y}_i) + \lambda_m \sum_i p_i c_i(\mathbf{y}) \quad (4) \end{aligned}$$

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

4 Related Work

In this section we compare our framework with previously proposed methods.

Chang et al. propose constraint-driven learning (CODL)(Chang et al., 2007). It can be interpreted as a variation of the self-training algorithm with constraints, where training data is picked based on the model's prediction and the constraints. By directly incorporating the model score and the constraints (as in Equation 4) we avoid the expensive "Top-K" inference step in CODL.

Generalized expectation (GE) criterion (Mann and McCallum, 2008) and Alternating Projections (AP) (Bellare et al., 2009) express preferences by specifying constraints on feature expectations, which require expensive inference. Even though AP introduces an online version, it still involves full inference over each instance. Furthermore, these methods are restricted by the features of the model, while our approach can use arbitrary constraints on the factor graph.

(Li, 2009) incorporates prior knowledge into conditional random fields using virtual evidence. Our constraints are not encoded as variables in the factor graph, allowing more expressivity.

5 Experiments

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

We use the objective functions defined in Section 3.3, 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 $\alpha = 1.0$, $\lambda_s = 10$, and $\lambda_m = 0.0001$.

Average token accuracy for 5 runs is reported and compared with CODL in Table 1. We also report supervised results from (Chang et al., 2007), and by SampleRank. All of our methods demonstrate vast improvement over the supervised method for smaller training sizes, but this difference reduces as the training size increases. When the complete training data is used, additional unlabeled data hurt our performance. This is not observed in CODL as it uses more unlabeled data, which may also explain their slightly higher accuracy. Note that

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

Self+Cons performs better than Self or Cons individually. Model+Cons also performs competitively, and may potentially outperform other methods with a different λ_m.

Self training took 90 minutes to run on average, while Self+Cons and Model+Cons took 100 minutes. Since the Cons method skips the inference step over unlabeled data, it took only 30 minutes to run. As the size of the factor graphs and unlabeled data set grows, this saving will become more significant.

6 Conclusion

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

In future we will investigate the framework futher. This work only explored linear-chain based models, however we feel that the method can benefit more for large complex factor graphs such as those used for joint inference over multiple extraction tasks. 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. 2009. Alternating projections for learning with expectation constraints. In *UAI*.

Mingwei Chang, Lev Ratinov, and Dan Roth. 2007. Guiding semi-supervision with constraint-driven learning. In *ACL*.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithm. In *ACL*.

Aron Culotta, Michael Wick, and Andrew McCallum. 2007. First-order probabilistic models for coreference resolution. In *NAACL: Human Language Technologies (NAACL/HLT)*.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *ICML*.

Xiao Li. 2009. On the use of virtual evidence in conditional random fields. In *EMNLP*.

Gideon S. Mann and Andrew McCallum. 2008. Generalized expectation criteria for semi-supervised learning of conditional random fields. In *ACL*.

Andrew McCallum, Karl Schultz, and Sameer Singh. 2009. Factorie: Probabilistic programming via imperatively defined factor graphs. In *NIPS*.

Khashayar Rohanimanesh, Michael Wick, and Andrew McCallum. 2009. Inference and learning in large factor graphs with a rank based objective. Technical Report UM-CS-2009-08, University of Massachusetts, Amherst.

Sameer Singh, Karl Schultz, and Andrew McCallum. 2009. Bi-directional joint inference for entity resolution and segmentation using imperatively-defined factor graphs. In *ECML/PKDD*.