Structured Output Learning with Indirect Supervision: Review

Paper Review for class requirements of CS546 at UIUC

Narender Gupta
University of Illinois at Urbana-Champaign
ngupta18@illinois.edu

ABSTRACT

In this paper, we discuss the work "Structured Output Learning with Indirect Supervision" by Ming-Wei Chang, Vivek Srikumar, Dan Goldwasser, Dan Roth which was published in ICML 2010 [3]. We discuss the problem setting, method used by the authors to solve the problem, and comparable contemporary approaches. We also discuss several experiments conducted by the authors and try to analyze their setting and their results. Also, we pose some questions which are relevant to the work in terms of theoretical bounds as well as associations of several components of the framework.

1. INTRODUCTION

The authors study the problem of structured prediction in machine learning applications. A structured prediction problem can be thought of a set of constituent problems for each input example, such that the decision of one constituent affects the decision of the other. Due to the dependency of decisions, and hence inherent complexity of problem modeling and explosion of solution search space, the amount of data needed to learn a solution of the problem is intuitively very large. Annotating this data is very difficult because of the amount and complexity of annotation because of dependencies.

The authors propose a solution to this problem of availability of structurally annotated data, by suggesting that usually any structural problem is accompanied by a binary problem for which it is relatively easy to obtain annotated data. Learning this binary supervised problem can help obtain a better solution for the original structured prediction problem. It is an interesting approach because there have been works where learning an accompanying structured problem helps learning of binary problem [1]. But using easily available binary supervision data to aid highly expensive structural supervision data is intuitively going to accelerate the solution of harder problems.

2. PROBLEM FORMULATION

Let $S = \{(x_i, h_i)\}_{i=1}^{l}$ denotes the direct structural supervision set consisting of l examples x_i and their corresponding structures h_i.
Similarly, let $B = \{(x_i, y_i)\}_{i=l+1}^{l+m}$ where $y_i \in \{1, -1\}$ denotes the binary supervision set, also referred to as *indirect supervision* set.
For brevity, $i \in S$ indicates $(x_i, h_i) \in S$, and $i \in B$ indicates $(x_i, y_i) \in B$. $B^+ and B^-$ denote the partition of B consisting of positive and negative instances of B respectively. If $H(x)$ denotes the set of all feasible structures for x, and $\phi(x, h)$ be a feature generation function, then the goal of learning in a structured output prediction task is to learn a weight vector w such that it, for every example $(x_i, h_i) \in S$ assigns the highest score to the correct element h_i of $H(x)$. i.e.

$$h_i = \arg \max_{h \in H(x)} w^T \phi(x_i, h) \tag{1}$$

3. LEARNING FRAMEWORK

The authors propose a learning framework which is based on the structural version of SVM, where goal of learning is to solve the following minimization problem:

$$\min_w \frac{||w||^2}{2} + C_1 \sum_{i \in S} L_S(x_i, h_i, w) \tag{2}$$

where $L_S(x_i, h_i, w)$ represents structural loss function.

Given that there is a binary problem associated with the structure prediction problem, the authors add another loss term in the optimization problem, corresponding to the binary loss. The updated objective function is then written as:

$$\min_w \frac{||w||^2}{2} + C_1 \sum_{i \in S} L_S(x_i, h_i, w) + C_2 \sum_{i \in B} L_B(x_i, h_i, w) \tag{3}$$

In this objective function, $L_S(x_i, h_i, w)$ can be substituted with any structural loss such as hinge loss, or squared hinge loss, or any other loss which takes into account the distance of structures. And $L_B(x_i, y_i, w)$ can be expressed as:

$$L_B(x_i, y_i, w) = l\left(1 - y_i \max_{h \in H(x)} (w^T \phi_B(x_i, h))\right) \tag{4}$$

Since the binary task helps optimize for a better w without directly providing any structural supervision, the framework

is referred to as JLIS (Joint Learning with Indirect Supervision). The authors provide an optimization algorithm which uses cutting plane strategy. It splits the binary loss part of objective function into positive and negative parts, substitutes the positive label part with an approximation of the exact function based on iteratively updated model variable values, and then optimizes the convex function thus obtained. They also provide convergence guarantee for the algorithm under the assumption that l is non-decreasing function.

4. COMPARISON WITH OTHER FRAMEWORKS

Here, we compare the JLIS framework with some contemporary frameworks which either try to solve similar problems, or solve other problems in a similar fashion.

4.1 Latent SSVM

The objective function in equation 3 is very similar to the one proposed by Yu et al where they learn structural SVM with latent variable [6].

JLIS Objective function is written as:

$$\min_w \frac{||w||^2}{2} + C_1 \sum_{i \in S} L_S(x_i, h_i, w) + C_2 \sum_{i \in B} L_B(x_i, h_i, w)$$

where

$$L_S(x_i, h_i, w) = l\left(\max_h(\Delta(h, h_i) - w^T \phi_{h_i,h}(x_i))\right) \quad (5)$$

and

$$\phi_{h_i,h}(x_i) = \phi(x_i, h_i) - \phi(x_i, h) \quad (6)$$

and Δ is the distance function between two structures.

Latent Structure SVM objective is written as:

$$\min_w \frac{||w||^2}{2} + C \sum_i \max_{(h,\delta) \in H \times D} [w^T \phi(x_i, h, \delta) + \Delta(h_i, h, \delta)] - C \sum_i \max_{\delta \in D} w^T \phi(x_i, h_i, \delta)$$

where δ is the latent variable which can assume a value from D.

If the latent variable value set $D = \{1, -1, 0\}$, where -1 and 1 correspond to the binary supervision corresponding to the example, and 0 corresponding to structural supervision, then the JLIS formulation boils down to latent SSVM formulation.

Consequently, both of the algorithms i.e. JLIS as well as latent SSVM optimize using the same algorithm in a similar way, replacing the concave part with an approximate function achieved through iteratively attained best value of structure and then optimizing the convex function.

Unfortunately JLIS work doesn't compare or contrast itself from latent SSVM work. It'd have been interesting to see authors' point of view about the two works. But nevertheless, there are certain pros and cons of each of these which are evident. While JLIS provides a framework which facilitates supervision in an indirect form using labels of associated-problem, it is restricted to a binary supervision. Latent SSVM, on the other hand, allows for more than 2 labels to be incorporated into the structured prediction problem which are not the final structure labels. The problem with the latent variables in latent SSVM is that they do not necessarily have a semantic interpretation. Based on the data, or the way to estimate variables, they may assume certain values which are contrary to the human intuition. At the same time, since the binary problem in JLIS uses labels which are available with the data, the binary problem and the labels can always be interpreted semantically. Since latent SSVM is a modeling technique which uses data likelihood to estimate weight vectors, it doesn't need extra supervision data or labels for the binary/multilabel intermediate task. While it goes in the favor of latent SSVM that it works with less supervised data, it is a matter of concern if the indirect supervision is available and yet latent SSVM cannot use the available gold binary labels.

4.2 Contrastive Estimation

There is another work by Smith et al, named CE(Contrastive Estimation), which is similar to the one discussed here [5]. In this case, authors do provide a comparison between JLIS and Contrastive Estimation. The authors claim that the relationship between "good" and "bad" examples is not clear in all situations and hence application of CE in such a case might not be intuitive. The same argument, however, can also be made for the JLIS case, where it may be hard to come up with a binary supervision task associated with the structure prediction task which is either intuitively related, or is empirically effective.

JLIS has the advantage that it doesn't need to look for all the structures but only the best one, while CE needs to marginalize over all possible ones. It means less computation as well as inference time for JLIS while the corresponding times for CE may be significantly higher. But it also means that CE can potentially find a better structure even without any structural supervision which might be hard for JLIS. In fact, such a case is shown in the JLIS paper itself about the experiment of finding POS tag on ambiguous words. CE achieves the accuracy of 74.7% while JLIS achieves 70.1%.

5. EXPERIMENTS

The authors conduct several experiments to study the empirical nature of JLIS framework. These experiments are all under the purview of natural language applications. We discuss these experiments one by one, interpret their results, and then try to understand their purpose and effectiveness.

5.1 Phonetic Transliteration Alignment

Given a source language named entity (NE) and a corresponding target language NE, the goal of Phonetic Transliteration Alignment is to find the best phonetic alignment between the character sequences of the two NEs. The companion binary classification problem is the task of determining whether two words from different languages correspond to the same underlying entity.

The dataset is 300 pairs of NEs across English-Hebrew, where 100 pairs are reserved for training, and 200 for testing. The size of data, being so small, is a little worrying. When looking at the results, the structural SVM (SSVM) algorithm

achieves 72.9 F1 with just 10 examples as supervision and no indirect binary supervision. At this point, one should probably ask if the problem is too easy. Nevertheless, adding binary supervision does improve the performance of model, and the F1 goes from 72.9 to 80.0 with the same amount of structured supervision. Adding very little amount of structural supervision outperforms these numbers, but the margin decreases as the amount of structured supervision increases. It is not clear why the authors decided not to use all of the available training data in any of the experiments conducted. Also, there is no information on how the train-test split/sampling is done.

5.2 Part-of-Speech Tagging

The setting of this experiment is that 1000 sentences are chosen from Wall Street Journal, and their corresponding POS tags are used for structural supervision. The tokens in these sentences are also used in another 2000 sentences which provide binary supervision in the form of boolean answer to the question that there is a set of POS tags for this sentence or not.

The way size of structured supervision set is mentioned in this experiment is non-conventional. Instead of treating every sentence as a structural supervision example, the count of tokens is treated as the size of structural supervision set. This number, assuming average size of sentence in each sample is uniform, is a rough representation of the supervision set, but not the exact representation. Assuming average number of tokens in a sentence to be 8-10, the value of $|S|$ will range from $[20, 200]$ instead of $[200, 1600]$. Again, achieving such high numbers of $F1(> 70\%)$ with such small supervision beg the question about the difficulty of the problem. Leaving that aside, the trend of increasing F1 with added structural supervision, and then diminishing improvements with further increase show similar trend, as in Section 5.1.

5.3 Information Extraction

Under this section, the authors perform the task of identifying predefined fields in text. Associated tasks are extracting fields from an academic publication citation, and extracting fields from an advertisement text. Associated binary supervision problem is whether the text is well-structured or not. For the citation experiment, 300 structured labelled examples are used for training, 100 for development, and 100 for testing. For the advertisement experiment, 100 examples are used for training, development, and testing. It is unclear whether overall 100 examples are used, or 100 are used for each of the training, development and testing. Again, the size of structural supervision mentioned in the experimental results is reported as a function of tokens, and not the examples, which is non-conventional and slightly misleading. Reporting training data size in terms of token hinders the comparison of structured v/s binary training data size because, as defined by the framework, there should be only one binary label corresponding to a set of token labels (structure).

Again, in both of these examples, similar to Section 5.1 and 5.2, very small amount of data size leads to decent results in terms of F1. And the increasing structured supervision shows similar trend in terms of improvement.

5.4 Common to all Experiments

There are certain patterns in all of the experiments conducted in Section 5.1, 5.2, and 5.3. These can very briefly be summarized as:

1. Size of problem (training/testing data) is too small.
2. Very little amount of structural supervision provides good results.
3. Ratio of data for binary to structural supervision is too high.

6. QUESTIONS AND EXTENSIONS

All of the observations in Section 5.4 question the empirical validity of the theoretical JLIS framework as the data size grows, or the problem becomes harder to solve.

6.1 Questions

Following questions stem about the framework and it's setting:

1. Can we say anything about the nature of relationship between the structural supervision and associated binary supervision problem?
2. Is there any theoretical relationship between the size of data for indirect supervision needed to attain certain performance for structural supervision task?
3. Is it possible to learn a structured prediction problem from only binary supervised data under some limits? As shown from the CE experimented conducted in paper, zero supervision allows learning of non-zero performance for structured prediction task. Also, there is trend of increasing performance in structured prediction with increase in indirect supervision. Combining both of these observations, one can hypothesize that it is possible to learn structure prediction task without structure supervision at all. But is there any theoretical proof or bound for such a claim? Or is it even possible even within the limit?

6.2 Extensions

As is intuitive and can be observed from the empirical experiments, performance of structured prediction task goes up with indirect supervision under JLIS. Here are some possible extensions of the framework.

1. **JLIS with constraints:** Chang et al have shown in their work related to Constrained Conditional Models that adding constraints to a structured prediction problem can help improve it's performance [2]. Adding constraints to JLIS framework is the next logical step since JLIS helps boost performance of a model through indirect supervision from a companion problem data. This data can also help find constraints for the domain of the problem, apart from the world and domain knowledge constraints, and then these constraints can be fed to the model. Such a model can be thought of as the JLIS model working as first half of CCM model equation, and the constraints as a guide towards optimal weight vector for the problem.

2. **JLIS in graphs:** Graphs can intuitively be modeled as structured problems where the network structure serves as the structure for prediction task. Similarly, problems in other areas can be modeled better by adapting to graph structure for underlying patterns such as nested mention extraction using mention hypergraphs by Lu et al [4]. Adapting JLIS to such applications could be highly beneficial. Such an approach could also help network science problems such as community detection or graph clustering for the purposes of data flow or graph compression for approximated inference.

7. CONCLUSIONS

This paper discusses JLIS framework for structured prediction problems which proposes joint learning by using indirect supervision. The key contribution of the work is proposal of a discriminative joint learning framework, which uses a companion binary problem to aid the learning of original structured learning problem. Since binary supervision is relatively cheaper, such an approach can be beneficial in cases where getting annotated structured data is either expensive or hard to obtain. The authors conduct several experiments to prove empirical effectiveness of the framework, and success to a certain level. It'd be interesting to see the framework working in a setting where the task is "harder" or the size of data is "larger". Also, convergence guarantee is provided for the proposed framework but error bounds or limits of data size are needed to provide theoretical guarantees from perspective of learning.

8. REFERENCES

[1] M.-W. Chang, D. Goldwasser, D. Roth, and Y. Tu. Unsupervised constraint driven learning for transliteration discovery. *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, (June):299–307, 2009.

[2] M.-w. Chang, L. Ratinov, N. Rizzolo, and D. Roth. Learning and Inference with Constraints. *Aaai*, pages 1513–1518, 2008.

[3] M.-w. Chang, V. Srikumar, D. Goldwasser, and D. Roth. Structured Output Learning with Indirect Supervision. *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 199–206, 2010.

[4] W. Lu and D. Roth. Joint mention extraction and classification with mention hypergraphs. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP2015)*, 2015.

[5] N. A. Smith and J. Eisner. Contrastive estimation: Training Log-Linear Models on Unlabeled Data. In *ACL'05 - 43rd Annual Meeting of the Association for Computational Linguistics*, pages 354–362, 2005.

[6] C. Yu and T. Joachims. Learning structural SVMs with latent variables. *... International Conference on Machine Learning*, pages 1–8, 2009.