Training Structured Prediction Energy Networks with Indirect Supervision

Amirmohammad Rooshenas, Aishwarya Kamath, Andrew McCallum
College of Information and Computer Sciences
University of Massachusetts Amherst
{pedram,akamath,mccallum}@cs.umass.edu

Abstract

This paper introduces rank-based training of structured prediction energy networks (SPENs). Our method samples from output structures using gradient descent and minimizes the ranking violation of the sampled structures with respect to a scalar scoring function defined using domain knowledge. We have successfully trained SPEN for citation field extraction without any labeled data instances, where the only source of supervision is a simple human-written scoring function. Such scoring functions are often easy to provide; the SPEN then furnishes an efficient structured prediction inference procedure.

1 Introduction

Structured prediction, or the task of predicting multiple inter-dependent variables, is important in many domains, including computer vision, computational biology and natural language processing. For example, in sequence labelling, image segmentation, and parsing we are given input variables $\mathbf{x}$, and must predict output variables $\mathbf{y}$, where the number of possible $\mathbf{y}$ values are typically exponential in the number of variables that comprise it. Not only does this sometimes give rise to computational difficulties, it also leads to statistical parameter estimation issues, where learning precise models requires large amounts of labeled training data.

In some cases, unsupervised learning from plentiful unlabeled data may provide helpful outputs (Daumé III, 2009; Ammar et al., 2014). But usually some form of more direct supervision is required to create a model truly useful to the task at hand. In the absence of abundant labeled data we may consider alternative forms of supervision. For example, rather than providing labeled data instances, humans may more easily inject their domain knowledge by providing "labels on features," or "expectations" about correct outputs, as in generalized expectation criteria (Mann and McCallum, 2010), or by providing constraints, as in posterior regularization (Ganchev et al., 2010) or constraint driven learning (Chang et al., 2007). A major weakness of these methods, however, is that at training time inference must be done in the factor graph encompassing the *union* of the model's factor graph and the expectation dependencies—often leading to prohibitively expensive inference. Moreover, these methods cannot learn from non-decomposable domain knowledge, where the domain knowledge is not in a form of a set of labeled features or constraints.

An easy way for humans to express domain knowledge is by writing a simple scalar scoring function that indicates preferences among choices for $\mathbf{y}$ given $\mathbf{x}$. These human-coded functions may, for example, be based on arbitrary rule systems (or even Turing-complete programs) of the sort written by humans to solve problems before machine learning became so wide-spread.

In general, the human written domain knowledge functions are not expected to be perfect—most likely only examining a subset of features and not covering all cases. Thus we are now faced with two challenges: (1) the domain knowledge functions have limited generalization; (2) the domain knowledge functions provide a ranking, but do not provide an inference (search) procedure.

This paper presents a new training method for structured prediction energy networks (SPENs) (Belanger and McCallum, 2016; Belanger et al., 2017) that aims to address both these challenges, yielding efficient inference for structured prediction, trained from human-coded domain knowledge plus unlabeled data, but not requiring any labeled data instances. In SPENs, the factor graph that typically represents

output variable dependencies is replaced with a deep neural network that takes $\mathbf{y}$ and $\mathbf{x}$ as input and outputs a scalar energy score, but is able to learn much richer correlations than are typically captured in factor graphs. Inference in SPENs is performed by gradient descent in the energy, back-propagated to cause steps in a relaxed $\mathbf{y}$ space. Whereas previous training procedures for SPENs used labeled data, here we train SPENs from only unlabeled data plus human-coded domain knowledge in the form of a scoring function. We do so by building on SampleRank (Rohanimanesh et al., 2011; Singh et al., 2010), which enforces that the rank of two sampled $\mathbf{y}$s according to the trained factor graph is consistent with their rank according to distance to the labeled, true $\mathbf{y}$. In our training method, pairs of $\mathbf{y}$'s are obtained from successive steps of training-time gradient-descent inference on $\mathbf{y}$; when their rank is not consistent with that of the domain knowledge function, we accordingly update the energy network parameters.

We demonstrate our method on a citation field extraction task, for which we learn a neural network (1) that generalizes beyond the original domain knowledge function, and (2) that provides efficient test-time inference by gradient descent.

2 Structured Prediction Energy Networks

In general, SPEN parameterizes an energy function $E_{\mathbf{w}}(\mathbf{y}, \mathbf{x})$ using deep neural networks over output variables $\mathbf{y}$ as well as input variables $\mathbf{x}$, where $\mathbf{w}$ denotes the neural network's parameters. Belanger and McCallum (2016) separate the energy function into global and local terms. The role of the local terms is to capture the dependency among input $\mathbf{x}$ and each individual output variable y_i, while the global term aims to capture long-range dependencies among output variables.

Prediction in SPENs requires finding $\hat{\mathbf{y}} = \operatorname{argmin}_{\mathbf{y} \in \mathcal{Y}} E_{\mathbf{w}}(\mathbf{y}, \mathbf{x})$ for the given input $\mathbf{x}$. This inference problem is solved using gradient descent. However, the energy surface is non-convex, which prevents gradient descent inference from finding the exact structure $\mathbf{y}_{min}$ that globally minimizes the energy function. One approach to address this problem is to parameterize the energy function such that the SPEN is convex in the output variables $\mathbf{y}$ (Amos et al., 2017), but this limits the representational power of SPENs. Although gradient descent inference does not guarantee an exact solution, it has successfully been used in several domains such as multi-label classification (Belanger and McCallum, 2016), image-segmentation (Gygli et al., 2017), and semantic role labeling (Belanger et al., 2017).

3 Rank-Based Training of SPENs

Different methods have been introduced for training SPENs: margin-based training (Belanger and McCallum, 2016), end-to-end learning (Belanger et al., 2017), and value matching (Gygli et al., 2017). Margin-based training enforces the energy of the ground truth structure to be lower than the energy of every incorrect structure by a margin, which is calculated as the Hamming loss between the two structures. End-to-end learning unrolls the energy minimization into a differentiable computation graph to output the predicted structure. It then trains the model by directly minimizing the loss between the predicted and ground-truth structures. Finally, the value matching approach trains SPENs such that the energy value matches the value of a given target function, such as the L_2 distance between the ground-truth and predicted structures.

All of these methods strongly depend on the existence of the ground truth values either as labeled data or as the value of a function applied to it. While dependence of the margin-based and end-to-end learning approaches on the labeled data is explicit, this dependency in the case of value-matching may not be obvious. In the absence of labeled data, we have to use the model's predictions instead, for training. These predictions are often incorrect, especially at early stages of training. As a result, value-matching training is constrained to match the score of these predictions with the value of the energy function defined by SPEN. This requires matching several incorrect structures for a given input, which hinders gradient descent inference from finding the exact solution by introducing many local optima. To address this problem, we use a ranking objective similar to SampleRank (Rohanimanesh et al., 2011) such that it preserves the optimum points of the score function.

In general, if SPEN ranks every pair of output structures identical to the score function, the optimum points of the score function match those of SPEN. However, forcing the ranking constraint for every pair of output structures is not tractable, so

we need to approximate it by sampling some candidate pairs. Given a score function $V(\mathbf{y}, \mathbf{x})$, we are able to rank every two consecutive candidate structures based on their score values. Consider two candidate output structures $\mathbf{y}_1$ and $\mathbf{y}_2$ for the given input $\mathbf{x}$. We define $\mathbf{y}_h$ and $\mathbf{y}_l$ based on the score function as the following:

$$\mathbf{y}_h = \underset{\mathbf{y} \in \{\mathbf{y}_1, \mathbf{y}_2\}}{\operatorname{argmax}} V(\mathbf{y}, \mathbf{x}),$$
$$\mathbf{y}_l = \underset{\mathbf{y} \in \{\mathbf{y}_1, \mathbf{y}_2\}}{\operatorname{argmin}} V(\mathbf{y}, \mathbf{x}). \quad (1)$$

We expect that these two structures have the same ranking with respect to $E_{\mathbf{w}}(., \mathbf{x})$, which can be described as: $\alpha(V(\mathbf{y}_h, \mathbf{x}) - V(\mathbf{y}_l, \mathbf{x})) < E_{\mathbf{w}}(\mathbf{y}_h, \mathbf{x}) - E_{\mathbf{w}}(\mathbf{y}_l, \mathbf{x})$, where α is a tunable positive scalar. Therefore, the rank-based objective minimizes the constraint violations:

$$\min_{\mathbf{w}} \sum_{\mathbf{x} \in \mathcal{D}} [\alpha(V(\mathbf{y}_h, \mathbf{x}) - V(\mathbf{y}_l, \mathbf{x})) - E_{\mathbf{w}}(\mathbf{y}_h, \mathbf{x}) + E_{\mathbf{w}}(\mathbf{y}_l, \mathbf{x})]_+ \quad (2)$$

$[.]_+$ is $max(., 0)$. Figure 1 shows a ranking violation for two structures $\mathbf{y}_1$ and $\mathbf{y}_2$ for a given x. The arrows indicate the direction of update over the energy surface. Note that we ignore the dependence of $\mathbf{y}$ on $\mathbf{w}$, which introduces approximation in the gradient of Eq. 2. For the supervised setting, Belanger et al. (2017) address this problem by unrolling the inference steps as an inference network and back-propagating through the inference network. We leave exploring similar approaches for rank-based training for future work. To compute Eq. 2, we need to find configurations $\mathbf{y}_i$ and $\mathbf{y}_j$ such that both are candidate solutions for $\operatorname{argmin}_{\mathbf{y} \in \mathcal{Y}} E_{\mathbf{w}}(\mathbf{y}, \mathbf{x})$. If not, the number of required samples would be exponential in $|\mathcal{Y}|$. Since at test time we use gradient descent inference, a similar method is used for generating candidate structures: the trajectory of points in the inference mechanism is used as the set of possible candidates. The idea of deterministic sampling from SPEN energy surface was first introduced by David Belanger (2017). We define the inference trajectory, $\mathcal{T}(\mathbf{x})$, as a sequence of output structures generated using projected gradient descent inference in order to find the minimum solution of $E_{\mathbf{w}}(., \mathbf{x})$.

Given a random initial structure $\mathbf{y}_0$, we define the inference trajectory as: $\mathcal{T}(\mathbf{x}) = \{\mathbf{y}_1, \cdots, \mathbf{y}_m\}$, where $\mathbf{y}_{t+1} = \mathcal{P}_{y \in \Delta_L}(\mathbf{y}_t - \eta \frac{\partial}{\partial \mathbf{y}} E_{\mathbf{w}}(\mathbf{y}_t, \mathbf{x}))$. $\mathcal{P}_{y \in \Delta_L}$ projects the values of $\mathbf{y}$ onto the probability simplex Δ_L over L values that each variable y can take. For each input $\mathbf{x}$ in the training data, we find the first consecutive structures $\mathbf{y}_i$, $\mathbf{y}_{i+1} \in \mathcal{T}(\mathbf{x})$ that violate the ranking constraint, then use Eq. 2 to reduce the number of violations. Algorithm 1 describes the complete training algorithm.

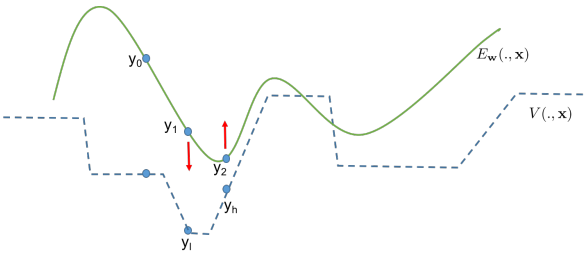

Figure 1: Schematic machinery of rank-based training. The dashed line is the surface of score function $V(., \mathbf{x})$ and the solid line is the surface of SPEN $E(., \mathbf{x})$, both conditioned on input $\mathbf{x}$. Here, $\mathbf{y}_2$ and $\mathbf{y}_3$ violate the ranking constraint, and the arrows show the direction of updates on the energy surface.

Algorithm 1 Rank-based training of SPEN

$\mathcal{D} \leftarrow$ unlabeled mini-batch of training data
$V(.,.) \leftarrow$ scoring function
$E_{\mathbf{w}}(.,.) \leftarrow$ input SPEN
for each $\mathbf{x}$ in $\mathcal{D}$ **do**
  $\mathcal{T}(\mathbf{x}) \leftarrow$ samples using GD inference in $E_{\mathbf{w}}(., \mathbf{x})$.
  $\xi \leftarrow \emptyset$.
  for each pair $(\mathbf{y}_i, \mathbf{y}_{i+1})$ in $\mathcal{T}(\mathbf{x})$ **do**
    Construct $\mathbf{y}_h$ and $\mathbf{y}_l$ using Eq.1
    if $\alpha(V(\mathbf{y}_h, \mathbf{x}) - V(\mathbf{y}_l, \mathbf{x})) > E_{\mathbf{w}}(\mathbf{y}_h, \mathbf{x}) - E_{\mathbf{w}}(\mathbf{y}_l, \mathbf{x})$ **then**
      $\xi \leftarrow \xi \cup (\mathbf{x}, \mathbf{y}_h, \mathbf{y}_l)$.
    end if
  end for
  Optimize Eq.2 using ξ.
end for

4 Citation Field Extraction

To show the success of rank-based learning with indirect supervision, we conduct experiments on citation field extraction as an instance of structured prediction problems. The goal of citation field extraction is to segment citation text into its constituent parts such as Author, Title, Journal, Page, and Date. We used the Cora citation dataset (Seymore et al., 1999), which includes 100 labeled examples as the test set and another 100 labeled examples for the validation set. Our training data consists of 300 training examples from the Cora citation data set for which we dismiss the labels,

as well as another 700 unlabeled citations acquired across the web, which adds up to 1000 unlabeled data points. Each token can be labeled with one of 13 possible tags. We use fixed-length input data by padding all citation text to the maximum citation length in the dataset, which is 118 tokens. We report token-level accuracy measured on non-pad tokens.

We provide the learning algorithm with a human written score function that takes the citation text and predicted tags as input. The score function then checks for violations of its rules and penalizes the predicted tags accordingly. Figure 2 shows examples of rules in the score function. Our complete score function consists of around 50 rules.

We used two 2-layer neural networks with 1000 and 500 hidden nodes to parameterize the local and global energy functions of SPEN. We examine different α (Eq. 2) values of 0.1, 1.0, 2.0, 5.0, and 10.0, and setting α value to 2.0 has the best performance on the validation set. We use gradient descent inference with ten gradient descent steps and $\eta = 0.1$ for both training and test.

We include the results of generalized expectation (GE) from Mann and McCallum (2010) that use the same dataset and setting. Our results show that R-SPEN achieves significantly better token-level accuracy as compared to GE.

We also compare R-SPEN with different inference algorithms that search using the score function to find the best configuration with maximum score. The results of these are listed in Table 1. Greedy search first randomly initializes the output tags and then iteratively replaces each assigned tag with a tag that results in the maximum score until the end of the citation is reached. This process is repeated until convergence, measured by no tag changing in an iteration. To avoid the effects of random initialization, this is repeated with varied number of random restarts, as reported in Table 1, where the best configuration is used in the scores reported. For the baseline that implements beam search, each citation is labeled by employing a beam search on the space of all tags for each token and their subsequent configurations, while keeping track of the best k configurations from one token to the next. This search is further augmented by restarting the search from the best k found after one complete search, for a total of 10 times and 10 random restarts.

```
score <- Contains the score of each example
first_seen <- Contains the index of
       the first appearance of each tag
j <- Index of the current token
i <- Index of the current example

# Parantheses have the same tag of what comes inside
if j > 0 and last_token == '(' or current_token == ')':
  if tags[j] != tags[j-1]:
    score[i] -= 1

# Period takes that tag of its predecessor
if j > 0 and current_token == '.':
  if j > 0 and tags[j] != tags[j-1]:
    score[i] -= 1

# Only one of the booktitle, journal,
#    or techinical report can appear
if first_seen[booktitle_tag] >= 0 :
  if first_seen[journal_tag] >= 0
      or first_seen[technical_report_tag] >= 0:
    score[i] -= 1

if first_seen[journal_tag] >= 0:
  if first_seen[booktitle_tag] >= 0
      or first_seen[technical_report_tag] >= 0:
    score[i] -= 1

if first_seen[techincal_report_tag] >= 0:
  if first_seen[booktitle_tag] >= 0
      or first_seen[journal_tag] >= 0:
    score[i] -= 1
```

Figure 2: Examples of rules in the score function. The first two rules constrain the relation of token and tags, while the last rule targets the relationship between tags.

Consulting Table 1, we can confirm that both greedy search and beam search find much better output structures in term of score values as compared to R-SPEN; however, they achieve poor accuracy because the domain knowledge function does not comprehensively provide rules regarding all possible output structures. We report the average score values of the R-SPEN predictions on test data as a function of training iterations in Figure 3. Within 1000 iterations, R-SPEN is able to achieve a test set accuracy of 38%, outperforming all baselines, while the average score is -18.0. R-SPEN generalizes beyond the domain knowledge function because it successfully captures the correlation among output variables through rank-based training on unlabeled data, so its predictions may have lower score values but are more accurate.

The test time inference of R-SPEN is much faster than search algorithms because SPEN provides efficient approximate inference.

5 Related Work

Generalized Expectation (GE) (Mann and McCallum, 2010), Posterior Regularization (Ganchev et al., 2010) and Constraint Driven Learning (Chang et al., 2007) are among well-known approaches to learn from domain knowledge decomposed over a set of constraints or labeled features. However, these methods cannot learn from black box domain knowledge based score functions. Score functions of this type are abundant in

Table 1: Comparison of R-SPEN with GE and different search algorithms in terms of token-level accuracy, test set average score, and time taken for inference during test time.

Method	Acc.	Avg. Score	Time (s)
GE	37.3%	N/A	—
Greedy Search			
10 restarts	22.6%	-4.92	700
100 restarts	26.0%	-3.26	6997
1000 restarts	26.1%	-2.51	69272
Beam Search			
k=2	30.0%	-1.87	4953
k=5	30.4%	-1.80	12217
k=10	31.0%	-1.44	22898
R-SPEN	47.1%	-20.33	<1

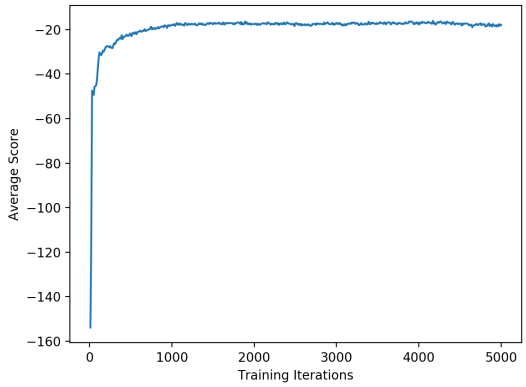

Figure 3: Average test set score values during training of R-SPEN.

various fields, for example, when the score is the result of evaluating a non-differentiable function over output structures.

Stewart and Ermon (2017) train a neural network using a score function that guides the training based on physics of moving objects. They have defined a differentiable score function which provides the learning algorithm with the gradient of the score function. However, in our approach the score function could be any complex non-differentiable function.

Peng et al. (2017) and Iyyer et al. (2017) use energy-based max-margin training for learning from an implicit source of supervision. This can be viewed as a score function evaluating the predicted output structure based on some real-world domain. For example, if the output structure is the SQL query associated with a natural language question, the score function can be specified as the Jaccard similarity of the extracted cells from the table using the generated SQL query and the set of gold answers for the natural language query as in Iyyer et al (2017).

6 Conclusion and Future Work

We have introduced a method to train structured prediction energy networks with indirect supervision that is derived from domain knowledge. This domain knowledge is a scalar function that is represented in the form of certain set of rules, easily provided by humans. By using a rank-based training we are able to effectively generalize beyond the domain knowledge function in problem instances where we do not have access to labeled data, thus establishing a viable option for solving structured prediction problems in those regimes.

R-SPEN only uses unlabeled data and domain knowledge for training. We should also effectively benefit from annotated data if any is available for the task. This can be accomplished by augmenting the domain knowledge with rules that take into account the distance between predicted and ground truth labels.

In the future, we wish to explore the effectiveness of R-SPEN on various tasks using domain knowledge functions with varying degrees of complexity.

Acknowledgments

This research was funded by DARPA grant FA8750-17-C-0106. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA or the U.S. Government.

References

Waleed Ammar, Chris Dyer, and Noah A Smith. 2014. Conditional random field autoencoders for unsupervised structured prediction. In *Advances in Neural Information Processing Systems*, pages 3311–3319.

Brandon Amos, Lei Xu, and J Zico Kolter. 2017. Input convex neural networks. In *Proceedings of the International Conference on Machine Learning*.

David Belanger. 2017. Deep energy-based models for structured prediction. *Ph.D. Dissertation*.

David Belanger and Andrew McCallum. 2016. Structured prediction energy networks. In *Proceedings of the International Conference on Machine Learning*.

David Belanger, Bishan Yang, and Andrew McCallum. 2017. End-to-end learning for structured prediction

energy networks. In *Proceedings of the International Conference on Machine Learning*.

Ming-Wei Chang, Lev Ratinov, and Dan Roth. 2007. Guiding semi-supervision with constraint-driven learning. In *ACL*, pages 280–287.

Hal Daumé III. 2009. Unsupervised search-based structured prediction. In *Proceedings of the International Conference on Machine Learning*, pages 209–216.

Kuzman Ganchev, Jennifer Gillenwater, Ben Taskar, et al. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11(Jul):2001–2049.

Michael Gygli, Mohammad Norouzi, and Anelia Angelova. 2017. Deep value networks learn to evaluate and iteratively refine structured outputs. In *Proceedings of the International Conference on Machine Learning*.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1821–1831.

Gideon S Mann and Andrew McCallum. 2010. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *Journal of machine learning research*, 11(Feb):955–984.

Haoruo Peng, Ming-Wei Chang, and Wen-tau Yih. 2017. Maximum margin reward networks for learning from explicit and implicit supervision. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2368–2378.

Khashayar Rohanimanesh, Kedar Bellare, Aron Culotta, Andrew McCallum, and Michael L Wick. 2011. Samplerank: Training factor graphs with atomic gradients. In *Proceedings of the International Conference on Machine Learning*, pages 777–784.

Kristie Seymore, Andrew McCallum, and Roni Rosenfeld. 1999. Learning hidden markov model structure for information extraction. In *AAAI-99 workshop on machine learning for information extraction*, pages 37–42.

Sameer Singh, Limin Yao, Sebastian Riedel, and Andrew McCallum. 2010. Constraint-driven rank-based learning for information extraction. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 729–732. Association for Computational Linguistics.

Russell Stewart and Stefano Ermon. 2017. Label-free supervision of neural networks with physics and domain knowledge. In *AAAI*, pages 2576–2582.