Learning Maximum-A-Posteriori Perturbation Models for Structured Prediction in Polynomial Time

Asish Ghoshal [1] **Jean Honorio** [1]

Abstract

MAP perturbation models have emerged as a powerful framework for inference in structured prediction. Such models provide a way to efficiently sample from the Gibbs distribution and facilitate predictions that are robust to random noise. In this paper, we propose a provably polynomial time randomized algorithm for learning the parameters of perturbed MAP predictors. Our approach is based on minimizing a novel Rademacher-based generalization bound on the expected loss of a perturbed MAP predictor, which can be computed in polynomial time. We obtain conditions under which our randomized learning algorithm can guarantee generalization to unseen examples.

1. Introduction

Structured prediction can be thought of as a generalization of binary classification to structured outputs, where the goal is to jointly predict several dependent variables. Predicting complex, structured data is of great significance in various application domains including computer vision (e.g., image segmentation, multiple object tracking), natural language processing (e.g., part-of-speech tagging, named entity recognition) and computational biology (e.g. protein structure prediction). However, unlike binary classification, structured prediction presents a set of unique computational and statistical challenges. The chief being that the number of structured outputs is exponential in the input size. For instance, in translation tasks, the number of parse trees of a sentence is exponential in the length of the sentence. Second, it is very common in such domains to have very few training examples as compared to the size of the output space thereby making generalization to unseen inputs difficult.

[1]Department of Computer Science, Purdue University, West Lafayette, IN - 47906. Correspondence to: Asish Ghoshal <aghoshal@purdue.edu>, Jean Honorio <jhonorio@purdue.edu>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

The key computational challenge in structured prediction stems from the *inference* problem, where a *decoder*, parameterized by a vector w of weights, predicts (or *decodes*) the latent structured output y given an observed input x. With the exception of a few special cases, the general inference problem in structured prediction is intractable. For instance in many cases the inference problem reduces to the maximum acyclic subgraph problem which is NP-hard and hard to approximate to within a factor of $1/2$ of the optimal solution (Guruswami et al., 2008), or cardinality-constrained submodular maximization, which is also NP-hard and hard to compute a solution better than the $(1 - 1/e)$-approximate solution returned by a greedy algorithm (Nemhauser et al., 1978). The *learning* problem, where the goal is to learn the parameter w of the decoder from a set of labeled training instances, and which involves solving the inference problem as a subroutine, is therefore intractable for all but a few special cases. Hardness of max-margin learning (SVM) was shown by (Sontag et al., 2010).

Hardness results notwithstanding, various methods — which are worst-case exponential-time — have been developed over the last decade for predicting structured data including conditional random fields (Lafferty et al., 2001), and max-margin approaches (Taskar et al., 2003), to name a few. In these approaches, learning the parameter w of the decoder involves minimizing a loss function $L(w, \mathsf{S})$ over a data set S of m training pairs $\{(x_i, y_i)\}_{i=1}^m$. One could also take a Bayesian approach and learn a posterior distribution $\mathcal{Q}$ over decoder parameters w by minimizing the Gibbs loss $\mathbb{E}_{w\sim\mathcal{Q}}[L(w, \mathsf{S})]$. McAllester (McAllester, 2007) showed, using the PAC-Bayesian framework, that the commonly used max-margin loss (Taskar et al., 2003) upper bounds the expected Gibbs loss over the data distribution, upto statistical error. Therefore, minimizing the max-margin loss provides a principled way for learning the parameters of a structured decoder. More recently, (Honorio & Jaakkola, 2016) showed that minimizing a *surrogate* randomized loss, where the max-margin loss is computed over a small number of randomly sampled structured outputs, also bounds the Gibbs loss from above upto statistical error.

The above can be thought of as weight based perturbation models. The perturb-and-MAP framework introduced by

(Papandreou & Yuille, 2011), and henceforth referred to as MAP perturbation, provides an efficient way to generate samples from the Gibbs distribution by injecting random noise (that do not depend on the weights of the decoder w) in the potential or score function of the decoder and then computing the most likely assignment or energy configuration (MAP). MAP perturbation models are an attractive alternative to expensive Markov Chain Monte Carlo simulations for drawing samples from the Gibbs distribution, in that the former facilitates one-shot sampling. Moreover, learning MAP predictors for structured prediction problems is particularly attractive because the predictions are robust to random noise. However, learning the parameters of such MAP predictors involves solving the MAP problem, which in general is intractable. In this paper we obtain a provably polynomial time algorithm for learning the parameters of perturbed MAP predictors with structure based perturbations. In the following paragraph we summarize the main technical contributions of our paper.

Our contributions. To the best of our knowledge, we are the first to obtain generalization bounds for MAP-perturbation models with structure-based (Gumbel) perturbations — for detailed comparison with existing literature see Section 6. While it is well known that Gumbel perturbations induce a conditional random field (CRF) distribution over the structured outputs, we show that the generalization error is upper bounded by a CRF loss up to statistical error. We obtain Rademacher based uniform convergence guarantees for the latter. However, the main contribution of our paper is to obtain a provably polynomial time algorithm for learning MAP-perturbation models for general structured prediction problems. We propose a novel randomized *surrogate loss* that lower bounds the CRF loss and still upper bounds the expected loss over data distribution, upto approximation and statistical error terms that decay as $\widetilde{\mathcal{O}}\left(1/\sqrt{m}\right)$ with m being the number of samples. While it is NP-Hard to compute and approximate the CRF loss in general (Barahona, 1982; Chandrasekaran et al., 2008), our surrogate loss can be computed in polynomial time. Our results also imply that one can learn parameters of CRF models for structured prediction in polynomial time under certain conditions. Our work is inspired by the work of (Honorio & Jaakkola, 2016) who also propose a polynomial time algorithm for learning the parameters of a structured decoder in the max-margin framework. In contrast to prior work which consider weight based perturbations, our work is concerned with structure based perturbations. Previous algorithms for learning MAP perturbation models, for instance, the hard-EM algorithm by (Gane et al., 2014) and the moment-matching algorithm by (Papandreou & Yuille, 2011), are in general intractable and have no generalization guarantees. *Lastly, the main conceptual contribution of our work is to demonstrate that it is possible to efficiently learn the parameters of a structured decoder with generalization guarantees without solving the inference problem exactly.*

2. Preliminaries

We begin this section by introducing our notations and formalizing the problem of learning MAP-perturbation models. In structured prediction, we have an input $x \in \mathfrak{X}$ and a set of feasible decodings of the input $\mathfrak{Y}(x)$. Without loss of generality, we assume that $|\mathfrak{Y}(x)| \leq r$ for all $x \in \mathfrak{X}$. Input-output pairs (x, y) are represented by a joint feature vector $\phi(x, y) \in \mathbb{R}^d$. For instance, when x is a sentence and y is a parse tree, the joint feature map $\phi(x, y)$ can be a vector of $0/1$-indicator variables representing if a particular word is present in x and a particular edge is present in y. We will assume that $\min\{\phi_j(x, y) \neq 0 \mid j \in [d]\} \geq 1$ which commonly holds for structured prediction problems, for instance, when using binary features, or features that "count" number of components, edges, parts, etc.

A decoder $f_w : \mathfrak{X} \to \mathfrak{Y}$, parameterized by a vector $w \in \mathbb{R}^d$, returns an output $y \in \mathfrak{Y}(x)$ given an input x. We consider linear decoders of the form:

$$f_w(x) = \underset{y \in \mathfrak{Y}(x)}{\operatorname{argmax}} \langle \phi(x, y), w \rangle, \tag{1}$$

which return the highest scoring structured output for a particular input x, where the score is linear in the weights w. As is traditionally the case in high-dimensional statistics, we will assume that the weight vectors are s-sparse, i.e., have at most s non-zero coordinates. We will denote the set of s-sparse d-dimensional vectors by $\mathbb{R}^{d,s}$.

In the perturb and MAP framework, a *stochastic decoder* first perturbs the linear score by injecting some independent noise for each structured output y, and then returns the structured output that maximizes the perturbed score. Gumbel perturbations are commonly used owing to the max-stability property of the Gumbel distribution. Denoting $\mathcal{G}(\beta)$ as the Gumbel distribution with location and scale parameters 0 and β respectively, we have the following stochastic decoder, where $\gamma \sim \mathcal{G}^r$ denotes a collection of r i.i.d. Gumbel-distributed random variables and γ_y denotes the Gumbel random variable associated with structured output y:

$$f_{w,\gamma}(x) = \underset{y \in \mathfrak{Y}(x)}{\operatorname{argmax}} \langle \phi(x, y), w \rangle + \gamma_y. \tag{2}$$

For any weight vector w, and data set $\mathsf{S} = \{(x_i, y_i)\} \overset{\text{i.i.d.}}{\sim} \mathcal{D}^m$, we consider the following expected and empirical zero-one loss:

$$L(w, \mathcal{D}) = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\mathbb{E}_{\gamma\sim\mathcal{G}^r} \left[\mathbf{1} \left[y \neq f_{w,\gamma}(x) \right] \right] \right], \tag{3}$$

$$L(w, \mathsf{S}) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_{\gamma\sim\mathcal{G}^r} \left[\mathbf{1} \left[y_i \neq f_{w,\gamma}(x_i) \right] \right], \tag{4}$$

where $\mathbf{1}[\cdot]$ denotes the indicator function and $\mathcal{D}$ is the unknown data distribution. We will let the scale parameter depend on the number of samples m and the weight vector w, and write $\beta(m, w) > 0$. The reason for this will become clear later, but intuitively one would expect that as the number of samples increases, the magnitude of perturbations should decrease in order to control the generalization error. Under Gumbel perturbations, $f_{w,\gamma}(x_i)$ is distributed according to following conditional random field (CRF) distribution $\mathcal{Q}(x_i, w)$ with pmf $q(\cdot; x_i, w)$ (Gumbel, 1954; Papandreou & Yuille, 2011):

$$q(y_i; x_i, w) = \Pr_{\gamma\sim\mathcal{G}^r(\beta)}\{f_{w,\gamma}(x_i) = y_i\} = \frac{\exp(\langle\phi(x_i,y_i),w\rangle/\beta)}{Z(w,x_i)}, \tag{5}$$

where $Z(w, x_i) = \sum_{y\in\mathcal{Y}(x)} \exp(\langle\phi(x_i,y),w\rangle/\beta)$ is the partition function. The empirical loss in (4) can then be computed as:

$$(\textbf{CRF loss})\ L(w,\mathsf{S}) = \frac{1}{m}\sum_{i=1}^{m}\Pr\{f_{w,\gamma}(x_i)\neq y_i\}. \tag{6}$$

The ultimate objective of a learning algorithm is to learn a weight vector w that generalizes to unseen data. Therefore, minimizing the expected loss given by (3) is the best strategy towards that end. However, since the data distribution is unknown, one instead minimizes the empirical loss (4) on a finite number of labeled examples S.

3. Generalization Bound

As a first step we will show that the empirical loss (6) indeed bounds the expected perturbed loss (3) from above, upto statistical error that decays as $\widetilde{\mathcal{O}}(1/\sqrt{m})$. We have the following generalization bound.

Theorem 1 (Rademacher based generalization bound)**.** *With probability at least $1-\delta$ over the choice of m samples S:*

$$(\forall w\in\mathbb{R}^{d,s})\ L(w,\mathcal{D}) \le L(w,\mathsf{S}) + \varepsilon(d,s,m,r,\delta),$$

where

$$\varepsilon(d,s,m,r,\delta) = 2\sqrt{\frac{s(\ln d + 2\ln(mr))}{m}} + 3\sqrt{\frac{\ln 2/\delta}{2m}}.$$

Proof. Let

$$g_w(x,y) \stackrel{\text{def}}{=} \Pr_{\gamma\sim\mathcal{G}^r(\beta)}\{y\neq f_{w,\gamma}(x)\},$$
$$\mathfrak{G} \stackrel{\text{def}}{=} \{g_w \mid w\in\mathbb{R}^{d,s}\}.$$

Then by Rademacher based uniform convergence, with probability at least $1-\delta$ over the choice of m samples, we have that:

$$(\forall w\in\mathbb{R}^{d,s})\ L(w,\mathcal{D}) \le L(w,\mathsf{S}) + 2\widehat{\mathfrak{R}}_{\mathsf{S}}(\mathfrak{G}) + 3\sqrt{\frac{\log 2/\delta}{2m}}, \tag{7}$$

where $\widehat{\mathfrak{R}}_{\mathsf{S}}(\mathfrak{G})$ denotes the empirical Rademacher complexity of $\mathfrak{G}$. Let $\sigma = (\sigma_i)_{i=1}^m$ be independent Rademacher variables. Also define $\mathcal{W} \stackrel{\text{def}}{=} \{w/\beta(w,m) \mid w\in\mathbb{R}^{d,s}\}$. Then,

$$\begin{aligned}
&\widehat{\mathfrak{R}}_{\mathsf{S}}(\mathfrak{G})\\
&= \mathbb{E}_\sigma\left[\sup_{w\in\mathbb{R}^{d,s}}\frac{1}{m}\sum_{i=1}^{m}\sigma_i g_w(x_i,y_i)\right]\\
&= \frac{1}{m}\mathbb{E}_\sigma\left[\sup_{w\in\mathbb{R}^{d,s}}\sum_{i=1}^{m}\sigma_i\Pr_{\gamma\sim\mathcal{G}^r(\beta)}\{y_i\neq f_{w,\gamma}(x_i)\}\right]\\
&\stackrel{(a)}{=} \frac{1}{m}\mathbb{E}_\sigma\left[\sup_{w\in\mathcal{W}}\sum_{i=1}^{m}\sigma_i\Pr_{\gamma\sim\mathcal{G}^r(1)}\{y_i\neq f_{w,\gamma}(x_i)\}\right]\\
&\le \frac{1}{m}\mathbb{E}_{\gamma\sim\mathcal{G}^r(1)}\left[\mathbb{E}_\sigma\left[\sup_{w\in\mathcal{W}}\sum_{i=1}^{m}\sigma_i\mathbf{1}[y_i\neq f_{w,\gamma}(x_i)]\right]\right]\\
&\stackrel{(b)}{\le} \frac{1}{m}\mathbb{E}_{\gamma\sim\mathcal{G}^r(1)}\left[\mathbb{E}_\sigma\left[\sup_{w\in\mathbb{R}^{d,s}}\sum_{i=1}^{m}\sigma_i\mathbf{1}[y_i\neq f_{w,\gamma}(x_i)]\right]\right],
\end{aligned}$$

where step (a) follows from $\Pr_{\gamma\sim\mathcal{G}^r(\beta)}\{y_i\neq f_{w,\gamma}(x_i)\} = \Pr_{\gamma\sim\mathcal{G}^r(1)}\{y_i\neq f_{w/\beta,\gamma}(x_i)\}$, and step (b) follows from $\mathcal{W}\subseteq\mathbb{R}^{d,s}$. We will enumerate the structured outputs $\mathcal{Y}(x_i)$ as $y_{i,1},\ldots,y_{i,r}$. For any fixed γ, the weight vector w induces a linear ordering $\pi_i(\cdot;\gamma)$ over the structured outputs $\mathcal{Y}(x_i)$, i.e., $\langle\phi(x_i,y_{i,\pi_i(1;\gamma)}),w\rangle + \gamma_1 > \langle\phi(x_i,y_{i,\pi_i(2;\gamma)}),w\rangle + \gamma_2 > \ldots > \langle\phi(x_i,y_{i,\pi_i(r;\gamma)}),w\rangle + \gamma_r$. Let $\pi(\gamma) = \{\pi_i\}$ be the orderings over all m data points induced by a fixed weight vector w and fixed γ, and let $\Pi(\gamma)$ be the collection of all orderings $\pi(\gamma)$ over all $w\in\mathbb{R}^{d,s}$ for a fixed γ. Since w is s-sparse we have, from results by (Bennett, 1956; Bennett & Hays, 1960; Cover, 1967), that the number of possible linear orderings is $|\Pi(\gamma)| \le \binom{d}{s}(mr)^{2s} \le d^s(mr)^{2s}$. Therefore we have:

$$\begin{aligned}
&\widehat{\mathfrak{R}}_{\mathsf{S}}(\mathfrak{G})\\
&\le \frac{1}{m}\mathbb{E}_{\gamma\sim\mathcal{G}^r(\beta)}\left[\mathbb{E}_\sigma\left[\sup_{\pi(\gamma)\in\Pi(\gamma)}\sum_{i=1}^{m}\sigma_i\mathbf{1}[y_i\neq y_{i,\pi_i(1;\gamma)}]\right]\right]\\
&\stackrel{(a)}{\le} \frac{1}{m}\sqrt{s(\log d + 2\log(mr))}\sqrt{m}\\
&= \sqrt{\frac{s(\log d + 2\log(mr))}{m}},
\end{aligned}$$

where the inequality (a) follows from the Massart's finite class lemma. □

As a direct consequence of the uniform convergence bound given by Theorem 1, we have that minimizing the CRF loss

(6) is a consistent procedure for learning MAP-perturbation models.

4. Towards an efficient learning algorithm

While Theorem 1 provides theoretical justification for learning MAP-perturbation models by minimizing the CRF loss (6), with the exception of a few special cases, computing the loss function is in general intractable. This is due to the need for computing the partition function $Z(w, x)$ which is an NP-hard problem (Barahona, 1982). Further, even approximating $Z(w, x)$ with high probability and arbitrary precision is also known to be NP-hard (Chandrasekaran et al., 2008).

To counter this computational bottleneck, we propose an efficient stochastic decoder that decodes over a randomly sampled set of structured outputs. To elaborate further, given some $x \in \mathfrak{X}$, let $\mathcal{R}(x, w)$ be some *proposal distribution*, parameterized by x and w, over the structured outputs $\mathfrak{Y}(x)$. We generate a set T' of n structured outputs sampled independently from the distribution $\mathcal{R}$ and define the following *efficient* stochastic decoder:

$$f_{w,\gamma,\mathsf{T}'}(x) = \operatorname*{argmax}_{y \in \mathsf{T}'} \langle \phi(x, y), w \rangle + \gamma_y. \tag{8}$$

Therefore $f_{w,\gamma,\mathsf{T}'}(x)$ is distributed according to the CRF distribution $\mathcal{Q}(x, w, \mathsf{T}')$ with pmf $q(\cdot; x, w, \mathsf{T}')$ and support on T' as follows:

$$\begin{aligned} q(y; x, w, \mathsf{T}') &= \Pr_{\gamma \sim \mathcal{G}^n} \left\{ f_{w,\gamma,\mathsf{T}'}(x) = y \right\} \\ &= \frac{\mathbf{1}\left[y \in \mathsf{T}'\right]}{Z_{w,x,\mathsf{T}'}} \exp(\langle \phi(x,y), w \rangle / \beta), \end{aligned}$$

where $Z_{w,x,\mathsf{T}'} = \sum_{y' \in \mathsf{T}'} \exp(\langle \phi(x,y'), w \rangle / \beta)$. Note that the partition function $Z_{w,x,\mathsf{T}'}$ can be computed in time linear in n, since $|\mathsf{T}'| = n$. Now, let $\mathsf{T} = \{\mathsf{T}_i \mid x_i \in \mathsf{S}\}$ be the collection of n structured outputs sampled for each x_i in the data set, from the product distribution $\mathcal{R}(\mathsf{S}, w) \stackrel{\text{def}}{=} \times_{i=1}^{m} (\mathcal{R}(x_i)^n)$. *Note that the distribution $\mathcal{R}(\mathsf{S}, w)$ does not depend on the $\{y_i\}$'s in S. We denote the distribution over the collection of sets $\{\mathsf{T}_i\}$ by $\mathcal{R}(\mathsf{S}, w)$ to keep the notation light.* Additionally, we consider proposal distributions $\mathcal{R}(x, w)$ that are equivalent upto linearly inducible orderings of the structured output.

Definition 1 (Equivalence of proposal distributions (Honorio & Jaakkola, 2016))**.** *For any $x \in \mathfrak{X}$, two proposal distributions $\mathcal{R}(x, w)$ and $\mathcal{R}(x, w')$, with probability mass functions $p(\cdot; x, w)$ and $p(\cdot; x, w')$, are equivalent if:*

$$\begin{aligned} &\forall y, y' \in \mathfrak{Y}(x) \ : \langle \phi(x, y), w \rangle \leq \langle \phi(x, y'), w \rangle \\ &\quad \textit{and } \langle \phi(x, y), w' \rangle \leq \langle \phi(x, y'), w' \rangle \\ &\iff \forall y \in \mathfrak{Y}(x) \ p(y; x, w) = p(y; x, w'). \end{aligned}$$

We then write $\mathcal{R}(x, w) \equiv \mathcal{R}(x, w') \equiv \mathcal{R}(x, \pi(x))$, where $\pi(x)$ is the linear ordering over $\mathfrak{Y}(x)$ induced by w (and w').

Intuitively speaking, the above definition requires proposal distributions to depend only on the orderings of the values $\langle \phi(x, y_1), w \rangle, \ldots, \langle \phi(x, y_r), w \rangle$ and not on the actual value of $\langle \phi(x, y_j), w \rangle$.

To obtain an efficient learning algorithm with generalization guarantees, we will use *augmented* sets $\bar{\mathsf{T}} = \{\bar{\mathsf{T}}_i\}_{i=1}^m$, where $\bar{\mathsf{T}}_i = \mathsf{T}_i \cup \{y_i\}$. Then, given a random collection of structured outputs T, we consider the following *augmented randomized* empirical loss for learning the parameters of the MAP-perturbation model:

$$L(w, \mathsf{S}, \bar{\mathsf{T}}) = \frac{1}{m} \sum_{i=1}^{m} \Pr_{\gamma \sim \mathcal{G}^n} \left\{ f_{w,\gamma,\bar{\mathsf{T}}_i}(x_i) \neq y_i \right\}. \tag{9}$$

As opposed to the loss function given by (6), the loss in (9) can be computed efficiently for small n. Our next result shows that the randomized augmented loss lower bounds the full CRF loss $L(w, \mathsf{S})$ as long as $\bar{\mathsf{T}}_i$ is a *set*, i.e., contains only unique elements.

Lemma 1. *For all data sets S, $\mathsf{T}_i \subseteq \mathfrak{Y}(x_i)$, and weight vectors w:*

$$\begin{aligned} &L(w, \mathsf{S}, \bar{\mathsf{T}}) - L(w, \mathsf{S}) = \\ &\quad - \frac{1}{m} \sum_{i=1}^{m} \Pr_{\gamma} \left\{ f_{w,\gamma,\bar{\mathsf{T}}_i}(x_i) = y_i \right\} \times \\ &\qquad \Pr_{\gamma} \left\{ f_{w,\gamma}(x_i) \in (\mathfrak{Y}(x_i) \setminus \bar{\mathsf{T}}_i) \right\} \leq 0 \end{aligned} \tag{10}$$

Proof. For any $x \in \mathfrak{X}$, $\mathsf{T} \subseteq \mathfrak{Y}(x)$, $y \in \mathsf{T}$ and weight vector w:

$$\begin{aligned} &\Pr_{\gamma} \left\{ f_{w,\gamma}(x) = y \right\} - \Pr_{\gamma} \left\{ f_{w,\gamma,\mathsf{T}}(x) = y \right\} \\ &= e^{\langle \phi(x,y), w \rangle} \left\{ \frac{Z(w, x, \mathsf{T}) - Z(w, x)}{Z(w, x) Z(w, x, \mathsf{T})} \right\} \\ &= \frac{e^{\langle \phi(x,y), w \rangle}}{Z(w, x, \mathsf{T})} \frac{1}{Z(w, x)} \left\{ - \sum_{y' \in \mathfrak{Y}(x) \setminus \mathsf{T}} e^{\langle \phi(x,y'), w \rangle} \right\} \\ &= -\Pr_{\gamma} \left\{ f_{w,\gamma,\mathsf{T}}(x) = y \right\} \Pr_{\gamma} \left\{ f_{w,\gamma}(x) \in \mathfrak{Y}(x) \setminus \mathsf{T} \right\}. \end{aligned}$$

Since by construction $y_i \in \bar{\mathsf{T}}_i$, the final claim follows. □

Remark 1. *If $\bar{\mathsf{T}}_i = \mathfrak{Y}(x_i)$ then $L(w, \mathsf{S}) = L(w, \mathsf{S}, \bar{\mathsf{T}}_i)$.*

Next, we will show that an algorithm that learns the parameter w of the MAP-perturbation model, by sampling a small number of structured outputs for each x_i and minimizing the empirical loss given by (9), generalizes under various choices of the proposal distribution $\mathcal{R}$. Our first step in that direction would be to obtain uniform convergence guarantees for the stochastic loss (9).

4.1. Generalization bound

To obtain our generalization bound, we decompose the difference $L(w,\mathsf{S}) - L(w,\mathsf{S},\bar{\mathsf{T}})$ as follows:

$$L(w,\mathsf{S}) - L(w,\mathsf{S},\bar{\mathsf{T}}) = A(w,\mathsf{S}) + B(w,\mathsf{S},\bar{\mathsf{T}}), \quad (11)$$

$$A(w,\mathsf{S}) = L(w,\mathsf{S}) - \mathbb{E}_{\mathsf{T}\sim\mathcal{R}(\mathsf{S})}\left[L(w,\mathsf{S},\bar{\mathsf{T}})\right], \quad (12)$$

$$B(w,\mathsf{S},\bar{\mathsf{T}}) = \mathbb{E}_{\mathsf{T}\sim\mathcal{R}(\mathsf{S})}\left[L(w,\mathsf{S},\bar{\mathsf{T}})\right] - L(w,\mathsf{S},\bar{\mathsf{T}}), \quad (13)$$

where $A(w,\mathsf{S})$ can be thought of as the approximation error due to using a small number of structured outputs T_i's instead of the full sets $\mathcal{Y}(x_i)$, while $B(w,\mathsf{S},\bar{\mathsf{T}})$ be is the statistical error. In what follows, we will bound each of these errors from above.

From Lemma 1 it is clear that the proposal distribution plays a crucial role in determining how far the surrogate loss $L(w,\mathsf{S},\bar{\mathsf{T}})$ is from the CRF loss $L(w,\mathsf{S})$. To bound the approximation error, we make the following assumption about the proposal distributions $\mathcal{R}(x,w)$.

Assumption 1. *For all $(x_i,y_i) \in \mathsf{S}$ and weight vectors $w \in \mathbb{R}^{d,s}$, the proposal distribution satisfies the following condition with probability at least $1 - {\|w\|_1}/{\sqrt{m}}$, for a constant $c \in [0,1]$:*

(i) $\mathsf{T}_i = \{y_i\}$ *if* $\forall\, y \neq y_i \langle\phi(x_i,y_i),w\rangle > \langle\phi(x_i,y),w\rangle$,

(ii) $\frac{1}{n}\sum_{y\in\mathsf{T}_i}\langle\phi(x_i,y),w\rangle \geq \langle\phi(x_i,y_i),w\rangle + c\,\|w\|_1$ *otherwise,*

where the probability is taken over the set T_i.

Intuitively, Assumption 1 states that, if y_i is not the highest scoring structure under w, then the proposal distribution should return structures $\mathsf{T} = \{y\}$ whose average score is an additive constant factor away from the score of the observed structure y_i with high probability. Otherwise, the proposal distribution should return the singleton set $\mathsf{T} = \{y_i\}$ with high probability. Note that Assumption 1 is in comparison much weaker than the low-norm assumption of (Honorio & Jaakkola, 2016), which requires that, in expectation, the norm of the difference between $\phi(x,y)$ and $\phi(x,y_i)$ (where y is sampled from the proposal distribution) should decay as ${1}/{\sqrt{m}}$. The following lemma bounds the approximation error from above.

Lemma 2 (Approximation Error)**.** *If the scale parameter of the Gumbel perturbations satisfies: $\beta \leq \min({\|w\|_1}/{\log m}, {w_{\min}}/{\log((r-1)(\sqrt{m}-1))})$ for all $w \neq 0$, and $n \geq m^{0.5-c}$, then under Assumption 1 $A(w,\mathsf{S}) \leq \varepsilon_1(m,n,w)$, where*

$$\varepsilon_1(m,n,w) \stackrel{\text{def}}{=} \frac{\|w\|_1}{\sqrt{m}} + \frac{1}{1+\sqrt{m}},$$

and $w_{\min} = \min\{|w_j| \mid |w_j| \neq 0, j \in [d]\}$.

Proof. Let $A_i(w,\mathsf{S}) \stackrel{\text{def}}{=} \Pr_{\gamma\sim\mathcal{G}(\beta)}\{f_{w,\gamma}(x_i) \neq y_i\} - \mathbb{E}_{\mathsf{T}_i}\left[\Pr_{\gamma\sim\mathcal{G}(\beta)}\left\{f_{w,\gamma,\bar{\mathsf{T}}_i}(x_i) \neq y_i\right\}\right]$ be the i-th term of $A(w,\mathsf{S})$. We will consider two cases.

Case I: y_i is strictly the highest scoring structure for x_i under w, i.e., $\forall y \neq y_i\ \langle\phi(x_i,y_i),w\rangle > \langle\phi(x_i,y),w\rangle$. First note that:

$$A_i(w,\mathsf{S}) \leq \Pr_{\gamma\sim\mathcal{G}(\beta)}\{f_{w,\gamma}(x_i) \neq y_i\}. \quad (14)$$

We will prove that $\Pr_{\gamma\sim\mathcal{G}(\beta)}\{f_{w,\gamma}(x_i) \neq y_i\} \leq {1}/{\sqrt{m}}$. Assume instead that $\Pr_{\gamma\sim\mathcal{G}(\beta)}\{f_{w,\gamma}(x_i) \neq y_i\} > {1}/{\sqrt{m}}$. Then

$$\sum_{y\neq y_i}(\sqrt{m}-1)e^{\langle\phi(x_i,y),w\rangle/\beta} > e^{\langle\phi(x_i,y_i),w\rangle/\beta}$$

Let $y' \in \mathcal{Y}(x_i)\setminus\{y_i\}$ be such that $\langle\phi(x_i,y'),w\rangle$ is maximized. Then, $(r-1)(\sqrt{m}-1)e^{\langle\phi(x_i,y'),w\rangle/\beta}$ upper bounds the left-hand side of the above equation. Taking log on both sides we get:

$$\beta > \frac{\langle\phi(x_i,y_i) - \phi(x_i,y'),w\rangle}{\log((r-1)(\sqrt{m}-1))}$$

Since y_i is the unique maximizer of the score $\langle\phi(x_i,y_i),w\rangle$, $\phi(x_i,y')$ and $\phi(x_i,y_i)$ must differ on at least one element in the support set of w. This implies, from above and the assumption that the minimum non-zero element of $\phi(x,y)$ is at least 1:

$$\beta > \frac{w_{\min}}{\log((r-1)(\sqrt{m}-1))},$$

which violates Assumption 1. Therefore from (14) we have that $A_i(w,\mathsf{S}) \leq {1}/{\sqrt{m}}$.

Case II: $\exists y \neq y_i : \langle\phi(x_i,y),w\rangle \geq \langle\phi(x_i,y_i),w\rangle$. Let $\Delta_i(y) \stackrel{\text{def}}{=} \phi(x_i,y) - \phi(x_i,y_i)$. In this case,

$$\begin{aligned} A_i(w,\mathsf{S}) &\stackrel{(a)}{\leq} \mathbb{E}_{\mathsf{T}_i}\left[\Pr_\gamma\left\{f_{w,\gamma,\bar{\mathsf{T}}_i}(x_i) = y_i\right\}\right] \\ &= \mathbb{E}_{\mathsf{T}_i}\left[\frac{\exp(\langle\phi(x_i,y_i),w\rangle/\beta)}{Z(w,x_i,\bar{\mathsf{T}}_i)}\right] \\ &\stackrel{(b)}{=} \mathbb{E}_{\mathsf{T}_i}\left[\frac{1}{1+\sum_{y\in\mathsf{T}_i}e^{\langle\Delta_i(y),w\rangle/\beta}}\right] \\ &\stackrel{(c)}{\leq} \mathbb{E}_{S_i}\left[\frac{1}{1+ne^{S_i/\beta}}\right], \end{aligned} \quad (15)$$

where we have defined $S_i \stackrel{\text{def}}{=} \frac{1}{n}\sum_{y\in\mathsf{T}_i}\langle\Delta_i(y),w\rangle$. In the above, in step (a) we dropped the term $\Pr_\gamma\{f_{w,\gamma}(x_i) = y_i\}$ to get an upper bound. Step (b) follows from dividing the numerator and denominator by $\exp(\langle\phi(x_i,y_i),w\rangle)$ and that $y_i \in \bar{\mathsf{T}}_i$. Step (c) follows from Jensen's inequality. Now,

$$\mathbb{E}_{S_i}\left[\frac{1}{1+ne^{S_i/\beta}}\right]$$

$$
\begin{aligned}
&= \mathbb{E}_{S_i}\left[\frac{1}{1+ne^{S_i/\beta}} \mid S_i \geq \frac{\|w\|_1}{2}\right] \Pr\left\{S_i \geq \frac{\|w\|_1}{2}\right\} \\
&\quad + \mathbb{E}_{S_i}\left[\frac{1}{1+ne^{S_i/\beta}} \mid S_i < \frac{\|w\|_1}{2}\right] \Pr\left\{S_i < \frac{\|w\|_1}{2}\right\} \\
&\overset{(a)}{\leq} \mathbb{E}_{S_i}\left[\frac{1}{1+ne^{S_i/\beta}} \mid S_i \geq \frac{\|w\|_1}{2}\right] + \frac{\|w\|_1}{\sqrt{m}} \\
&\overset{(b)}{\leq} \mathbb{E}_{S_i}\left[\frac{1}{1+ne^{S_i \log m/\|w\|_1}} \mid S_i \geq \frac{\|w\|_1}{2}\right] + \frac{\|w\|_1}{\sqrt{m}} \\
&= \mathbb{E}_{S_i}\left[\frac{1}{1+nm^{S_i/\|w\|_1}} \mid S_i \geq \frac{\|w\|_1}{2}\right] + \frac{\|w\|_1}{\sqrt{m}} \\
&\leq \frac{1}{1+n\sqrt{m}} + \frac{\|w\|_1}{\sqrt{m}}, \qquad (16)
\end{aligned}
$$

where inequality (a) follows from Assumption 1 and (b) follows from the fact that $\beta \leq \|w\|_1/\log m$. Thus from (15) and (16) we have that $A_i(w, \mathsf{S}) \leq 1/(1+n\sqrt{m}) + \|w\|_1/\sqrt{m}$.

The final claim follows from Case I and II. □

Note that for $c \geq 0.5$ the number of structured outputs needed is $n = 1$, while in the worst case ($c = 0$) $n = \sqrt{m}$. Furthermore, n needs to grow polynomially with respect to m in order to achieve $\mathcal{O}(1/\sqrt{m})$ generalization error.

Lemma 3 (Statistical Error)**.** *For any fixed data set* S*, the statistical error* $B(w, \mathsf{S}, \bar{\mathsf{T}})$ *is bounded, simultaneously for all proposal distributions* $\mathcal{R}(x_i, w)$ *over* $\{\mathsf{T}_i\}$*, as follows:*

$$
\begin{aligned}
&\Pr_{\mathsf{T}}\left\{(\forall w \in \mathbb{R}^{d,s})\ B(w, \mathsf{S}, \bar{\mathsf{T}}) \leq \varepsilon_2(d, s, n, r, m, \delta) \mid \mathsf{S}\right\} \\
&\qquad \geq 1 - \delta, \qquad (17)
\end{aligned}
$$

where

$$
\varepsilon_2(d, s, n, r, m, \delta) \overset{\text{def}}{=} 2\sqrt{\frac{s(\ln d + 2\ln(nr))}{m}} + \sqrt{\frac{\ln 1/\delta}{2m}} + \sqrt{\frac{s(\ln d + 2\ln(mr)) + \ln 1/\delta}{2m}}.
$$

The proof of the above lemma is adapted from the proof of Rademacher based uniform convergence, and can be found in Appendix A in the supplementary material.

Now, we are ready to present our main result proving uniform convergence of the randomized loss $L(w, \mathsf{S}, \bar{\mathsf{T}})$. More specifically, we provide $\widetilde{\mathcal{O}}(1/\sqrt{m})$ generalization error.

Theorem 2. *With probability at least* $1 - 2\delta$ *over the choice of the data set* S *and the set of random structured outputs* T*, and simultaneously for all* $w \in \mathbb{R}^{d,s}$ *and proposal distributions* $\mathcal{R}(x, w)$*:*

$$
L(w, \mathcal{D}) \leq L(w, \mathsf{S}, \bar{\mathsf{T}}) + \varepsilon_1 + \varepsilon_2, \qquad (18)
$$

where ε_1 *and* ε_2 *are defined in Lemma 2 and 3 respectively.*

Proof. The claim follows directly from Lemma 2 and Lemma 3 by taking an expectation with respect to S. □

4.2. Examples of proposal distributions

Having proved uniform convergence of our randomized procedure for learning the parameters of a MAP decoder, we turn our attention to the proposal distribution. We want to construct proposal distributions of the form given by Definition 1 that satisfy Assumption 1 with a large enough constant c. Additionally, for our randomized procedure to run in polynomial time we want the proposal distribution to sample a structured output in constant time. The following algorithm is directly motivated by Assumption 1 where the set $\text{neighbors}_k(y)$ for an input x is defined as: $\text{neighbors}_k(y) \overset{\text{def}}{=} \{y' \in \mathfrak{Y}(x) \setminus \{y\} \mid H(y, y') \leq k\}$, with $H(\cdot, \cdot)$ being the Hamming distance.

Algorithm 1 An example algorithm implementing a proposal distribution that depends on $y_i \in \mathsf{S}$.

Input: Weight vector $w \in \mathbb{R}^{d,s}$, $(x_i, y_i) \in \mathsf{S}$, parameter $\alpha \in [0, 1]$ and $k \geq 1$.
Output: A structured output $y \in \mathfrak{Y}(x)$.
With probability α pick y' uniformly at random from $\mathfrak{Y}(x_i)$, and with probability $1 - \alpha$ set y' to y_i.
$y \leftarrow y'$.
for $y' \in \mathsf{neighbors}_k(y)$ **do**
  if $\langle \phi(x, y'), w \rangle \geq \langle \phi(x, y), w \rangle$ **then**
    $y \leftarrow y'$.
  end if
end for
Return y.

Remark 2. *Setting* $\alpha = \|w\|_1/\sqrt{m}$*, Algorithm 1 satisfies the condition given in Definition 1 as well as Assumption 1. Since, for any* $w, w' \in \mathbb{R}^{d,s}$ *that induce the same linear ordering over* $\mathfrak{Y}(x)$*, conditioned on the* y' *sampled in Step 3, the algorithm returns the same* y *for both* w *and* w' *with probability* 1.

Also note that using a larger k ensures that the above algorithm satisfies Assumption 1 with a larger constant c, thereby reducing the number of structured outputs that need to be sampled (n), at the cost of increased computation for sampling a single structured output.

The parameter α in Algorithm 1 controls exploration vs exploitation. As α becomes smaller Algorithm 1 returns a proposal from within the neighborhood of y_i while for larger α it explores high scoring structures in the entire set of candidate structures.

Lastly, note that our results do not violate the hardness results of (Sontag et al., 2010), who essentially show that it is NP-hard to decide if the training data is linearly separable. Depending on whether or not the data is linearly separable, the loss $L(w, S)$ (6) can be large or small (for all or some weight vector). While computing $L(w, S)$ is intractable in

general, we merely provide an efficiently computable lower bound $L(w, S, \mathsf{T})$ ((9)) that still upper bounds the expected loss $L(w, \mathcal{D})$.

4.3. Minimizing the CRF loss

In this section we discuss strategies for minimizing the (randomized) CRF loss $L(w, \mathsf{S}, \bar{\mathsf{T}})$. Minimizing the randomized CRF loss $L(w, \mathsf{S}, \bar{\mathsf{T}})$ is equivalent to maximizing the *randomized CRF gain* $U(w, \mathsf{S}, \bar{\mathsf{T}}) = \frac{1}{m}\sum_{i=1}^{m} \Pr_\gamma \left\{ f_{w,\gamma,\bar{\mathsf{T}}_i}(x_i) = y_i \right\}$, which in turn is equivalent to maximizing $\log U(w, \mathsf{S}, \bar{\mathsf{T}})$. The latter can be accomplished by gradient based methods with the gradient of $\log U(w, \mathsf{S}, \bar{\mathsf{T}})$ given by:

$$\nabla_w \log U(w, \mathsf{S}, \bar{\mathsf{T}}) = \frac{\sum_{i=1}^{m} q_i (\phi(x_i, y_i) - \mathbb{E}\left[\phi(x_i, y)\right])}{\sum_{i=1}^{m} q_i}, \tag{19}$$

where $q_i \stackrel{\text{def}}{=} \Pr_\gamma \left\{ f_{w,\gamma,\bar{\mathsf{T}}_i}(x_i) = y_i \right\}$, and the expectation is taken with respect to $y \sim \mathcal{Q}(x_i, w, \bar{\mathsf{T}}_i)$. The exact CRF loss ($L(w, \mathsf{S})$) can similarly be minimized by using $\bar{\mathsf{T}}_i = \mathfrak{Y}(x_i)$, for all $x_i \in \mathsf{S}$, in the above. Note that by Jensen's inequality $\log U(w, \mathsf{S}, \bar{\mathsf{T}}) \geq \frac{1}{m}\sum_{i=1}^{m} \log \Pr_\gamma \left\{ f_{w,\gamma,\bar{\mathsf{T}}_i}(x_i) = y_i \right\}$, where the latter can be identified as the log likelihood of the data set S under the CRF distributions $\{\mathcal{Q}(x_i, w, \bar{\mathsf{T}}_i)\}$. Therefore, $L(w, \mathsf{S}, \bar{\mathsf{T}})$ can be equivalently minimized by minimizing the negative log-likelihood of the data, which in turn gives rise to the well known *moment-matching* rule known in the literature (Papandreou & Yuille, 2011). Thus, Algorithm 1 can be used with standard moment matching where the expectation is approximated by averaging over y's drawn from the distribution $\mathcal{Q}(x_i, w, \bar{\mathsf{T}}_i)$. While standard moment matching is in general intractable, moment matching in conjunction with Algorithm 1 is always efficient. Indeed, (19) can be thought of as a "weighted" moment matching rule with weights q_i.

5. Experiments

In this section, we evaluate our proposed method (`CRF_RAND`) on synthetic data against three other methods: `CRF_ALL`, `SVM_RAND`, and `SVM`. The `CRF_RAND` method minimizes the randomized loss $L(w, \mathsf{S}, \bar{\mathsf{T}})$ (9) subject to ℓ_1 penalty (as prescribed by Lemma 2) by sampling structured outputs from the proposal distribution given by Algorithm 1. The `CRF_ALL` method minimizes the exact (exponential-time) loss $L(w, \mathsf{S})$ (6). Lastly, `SVM` is the widely used max-margin method of (Taskar et al., 2003), while `SVM_RAND` is the randomized SVM method proposed by (Honorio & Jaakkola, 2016).

We generate a ground truth parameter $w^* \in \mathbb{R}^d$ with random entries sampled independently from a zero mean Gaussian distribution with variance 100. We then randomly set all but $s = \sqrt{d}$ entries to be zero. We then generate a training set of S of 100 samples. We used the following joint feature map $\phi(x, y)$ for an input output pair. For every pair of possible edges or elements i and j, we set $\phi(x, y)_{i,j} = \mathbf{1}\left[x_{i,j} = 1 \wedge i \in y \wedge j \in y\right]$. For instance, for directed spanning trees of v nodes, we have $x \in \{0, 1\}^{\binom{v}{2}}$ and $\phi(x, y) \in \mathbb{R}^{\binom{v}{2}}$. We considered directed spanning trees of 6 nodes, directed acyclic graphs of 5 nodes and 2 parents per node, and sets of 4 elements chosen from 15 possible elements. In order to generate each training sample $(x, y) \in S$, we generated a random vector x with independent Bernoulli entries with parameter $1/2$. After generating x, we set $y = f_{w^*}(x)$, i.e., we solved (1) in order to produce the latent structured output y from the observed input x and the parameter w^*.

We set the ℓ_1 regularization parameter to be 0.01 for all methods. We used 20 iterations of gradient descent with step size of $1/\sqrt{t}$ for all algorithms, where t is the iteration, to learn the parameter w for both the exact method and our randomized algorithm. In order to simplify gradient calculations, we simply set $\beta = 1/\log((r-1))(\sqrt{m}-1))$ during training. For `CRF_RAND`, we used Algorithm 1 with $\alpha = \|w\|_1/\sqrt{m}$ and invoke the algorithm $\sqrt{m}$ number of times to generate the set T_i for each $i \in [m]$ and w. This results in $n = |\mathsf{T}_i| \leq \sqrt{m}$. To evaluate the generalization performance of our algorithm we generated a test set $S' = \{x'_i, y'_i\}_{i=1}^{m}$ of 100 samples and calculated two losses. The first was the full CRF loss (6) on the test set S', and the second was the test set hamming loss $\frac{1}{m}\sum_{i=1}^{m} \hat{H}(f_{\hat{w}}(x'_i), y'_i)$, where $\hat{H}(\cdot, \cdot)$ is the normalized Hamming distance, and $\hat{w}$ is the learned parameter. Hamming distance is a popular distortion function used in structured prediction, and provides a more realistic assessment of the performance of a decoder, since in most cases it suffices to recover most of the structure rather than predicting the structure exactly. For DAGs and trees the Hamming distance counts the number of different edges between the structured outputs, while for sets it counts the number of different elements. We normalize the Hamming distance to be between 0 and 1. We computed the mean and 95% confidence intervals of each of these metrics by repeating the above procedure 30 times.

Figure 1 shows the training and test set errors and the training time of the four different algorithms. `CRF_RAND` significantly outperformed other algorithms in both the test set loss and test set hamming loss, while being ≈ 6 times faster than the exact method (`CRF_ALL`) for DAGs, ≈ 20 times faster for trees, and ≈ 3 times faster for sets. The exact CRF method (`CRF_ALL`) was also significantly faster than the exact SVM (`SVM`) method while achieving similar test set loss and test set hamming loss.

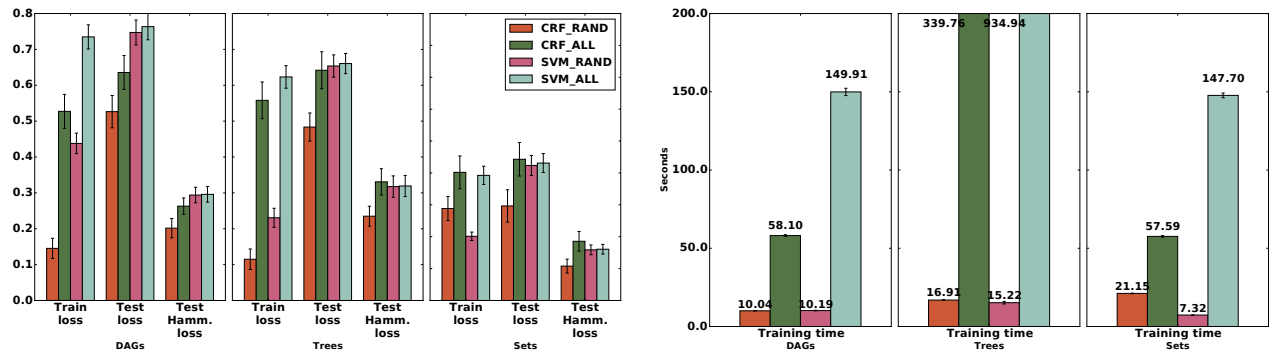

Figure 1. (**Left**) Training and test set loss (6), and test set hamming loss of the exact method (CRF_ALL) and our randomized algorithm (CRF_RAND), the randomized SVM method by (Honorio & Jaakkola, 2016) (SVM_RAND), and the exact SVM (SVM_ALL), a.k.a max-margin, method of (Taskar et al., 2003). For the randomized algorithms, i.e., CRF_RAND and SVM_RAND, the training loss is the randomized training loss, i.e., $L(w, \mathsf{S}, \bar{\mathsf{T}})$ and $L(w, \mathsf{S}, \mathsf{T})$ respectively. (**Right**) Training time in seconds for the various methods.

6. Related Work

Significant body of work exists in computing a single MAP estimate by exploiting problem specific structure, for instance, super-modularity, linear programming relaxations to name a few. However, in this paper we are concerned with the problem of learning the parameters of MAP perturbation models. Among generalization bounds for MAP perturbation models, (Hazan et al., 2013b) prove PAC-Bayesian generalization bounds for weight based perturbations. (Hazan et al., 2013b) additionally propose learning weight based MAP-perturbation models by minimizing the PAC-Bayesian upper bound on the generalization error. However, their method for learning the parameters involves constructing restricted families of posterior distributions over the weights w that lead to smooth, but not necessarily convex, generalization bounds that can be optimized using gradient based methods. For learning MAP-perturbation models with structure based (Gumbel) perturbations, (Gane et al., 2014) propose a hard-EM algorithm which is both worst-case exponential time and has no theoretical guarantees. (Papandreou & Yuille, 2011) on the other hand, propose learning Gumbel MAP-perturbation models by using the moment matching method. However, such an approach is tractable only for energy functions for which the global minimum can be computed efficiently. Lastly, (Hazan et al., 2013a; Orabona et al., 2014) consider the problem of efficiently sampling from MAP perturbation models using low dimensional perturbations. (Hazan & Jaakkola, 2012; Hazan et al., 2013a) additionally propose ways to approximate and bound the partition function. While such bounds on the partition function can be used, in principle, to approximately minimize the CRF loss (6), it is unclear if one can obtain uniform convergence guarantees for the same, given that computing or even approximating the partition function is NP-hard (Barahona, 1982; Chandrasekaran et al., 2008).

7. Concluding remarks

We conclude with some directions for future work. While in this work we showed that one can learn with approximate inference, it would be interesting to analyze approximate inference for prediction on an independent test set. Another avenue for future work would be to develop more powerful proposal distributions that allow for more finer-grained control over the parameter c by exploiting problem specific structure like submodularity.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1716609-IIS.

References

Barahona, F. On the computational complexity of ising spin glass models. *Journal of Physics A: Mathematical and General*, 15(10):3241, 1982.

Bennett, J. F. Determination of the number of independent parameters of a score matrix from the examination of rank orders. *Psychometrika*, 21(4):383–393, 1956.

Bennett, J. F. and Hays, W. L. Multidimensional unfolding: Determining the dimensionality of ranked preference data. *Psychometrika*, 25(1):27–43, 1960.

Chandrasekaran, V., Srebro, N., and Harsha, P. Complexity of inference in graphical models. In *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, pp. 70–78. AUAI Press, 2008.

Cover, T. M. The number of linearly inducible orderings of points in d-space. *SIAM Journal on Applied Mathematics*, 15(2):434–439, 1967.

Gane, A., Hazan, T., and Jaakkola, T. Learning with maximum a-posteriori perturbation models. In *Artificial Intelligence and Statistics*, pp. 247–256, 2014.

Gumbel, E. J. Statistical theory of extreme valuse and some practical applications. *Nat. Bur. Standards Appl. Math. Ser. 33*, 1954.

Guruswami, V., Manokaran, R., and Raghavendra, P. Beating the random ordering is hard: Inapproximability of maximum acyclic subgraph. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pp. 573–582. IEEE, 2008.

Hazan, T. and Jaakkola, T. On the partition function and random maximum a-posteriori perturbations. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pp. 1667–1674. Omnipress, 2012.

Hazan, T., Maji, S., and Jaakkola, T. On sampling from the gibbs distribution with random maximum a-posteriori perturbations. In *Advances in Neural Information Processing Systems*, pp. 1268–1276, 2013a.

Hazan, T., Maji, S., Keshet, J., and Jaakkola, T. Learning efficient random maximum a-posteriori predictors with non-decomposable loss functions. In *Advances in Neural Information Processing Systems*, pp. 1887–1895, 2013b.

Honorio, J. and Jaakkola, T. Structured prediction: from gaussian perturbations to linear-time principled algorithms. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pp. 271–278. AUAI Press, 2016.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pp. 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1. URL http://dl.acm.org/citation.cfm?id=645530.655813.

McAllester, D. Generalization bounds and consistency. *Predicting structured data*, pp. 247–261, 2007. URL http://nagoya.uchicago.edu/~dmcallester/colbounds.pdf.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978.

Orabona, F., Hazan, T., Sarwate, A., and Jaakkola, T. On measure concentration of random maximum a-posteriori perturbations. In *International Conference on Machine Learning*, pp. 432–440, 2014.

Papandreou, G. and Yuille, A. L. Perturb-and-map random fields: Using discrete optimization to learn and sample from energy models. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 193–200. IEEE, 2011.

Sontag, D., Meshi, O., Globerson, A., and Jaakkola, T. S. More data means less inference: A pseudo-max approach to structured learning. In *Advances in Neural Information Processing Systems*, pp. 2181–2189, 2010.

Taskar, B., Guestrin, C., and Koller, D. Max-margin Markov Networks. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'03, pp. 25–32, Cambridge, MA, USA, 2003. MIT Press. URL http://dl.acm.org/citation.cfm?id=2981345.2981349.