Structured Prediction: From Gaussian Perturbations to Linear-Time Principled Algorithms

Jean Honorio
CS, Purdue
West Lafayette, IN 47907, USA
jhonorio@purdue.edu

Tommi Jaakkola
CSAIL, MIT
Cambridge, MA 02139, USA
tommi@csail.mit.edu

Abstract

Margin-based structured prediction commonly uses a maximum loss over all possible structured outputs (Altun & Hofmann, 2003; Collins, 2004; Taskar et al., 2003). In natural language processing, recent work (Zhang et al., 2014; Zhang et al., 2015) has proposed the use of the maximum loss over random structured outputs sampled independently from some proposal distribution. This method is linear-time in the number of random structured outputs and trivially parallelizable. We study this family of loss functions in the PAC-Bayes framework under Gaussian perturbations (McAllester, 2007). Under some technical conditions and up to statistical accuracy, we show that this family of loss functions produces a tighter upper bound of the Gibbs decoder distortion than commonly used methods. Thus, using the maximum loss over random structured outputs is a principled way of learning the parameter of structured prediction models. Besides explaining the experimental success of (Zhang et al., 2014; Zhang et al., 2015), our theoretical results show that more general techniques are possible.

1 INTRODUCTION

Structured prediction has been shown to be useful in many diverse domains. Application areas include natural language processing (e.g., named entity recognition, part-of-speech tagging, dependency parsing), computer vision (e.g., image segmentation, multiple object tracking), speech (e.g., text-to-speech mapping) and computational biology (e.g., protein structure prediction).

In dependency parsing, for instance, the observed input is a sentence and the desired structured output is a parse tree for the given sentence.

In general, structured prediction can be viewed as a kind of decoding. A *decoder* is a machine for predicting the structured output y given the observed input x. Such a decoder, depends on a parameter w. Given a fixed w, the task performed by the decoder is called *inference*. In this paper, we focus on the problem of learning the parameter w. Next, we introduce the problem and our main contributions.

We assume a distribution D on pairs (x, y) where $x \in \mathcal{X}$ is the observed input and $y \in \mathcal{Y}$ is the latent structured output, i.e., $(x, y) \sim D$. We also assume that we have a training set S of n i.i.d. samples drawn from the distribution D, i.e., $S \sim D^n$, and thus $|S| = n$.

We let $\mathcal{Y}(x) \neq \emptyset$ denote the countable set of feasible *decodings* of x. In general, $|\mathcal{Y}(x)|$ is exponential with respect to the input size.

We assume a fixed mapping ϕ from pairs to feature vectors, i.e., for any pair (x, y) we have the feature vector $\phi(x, y) \in \mathbb{R}^k \setminus \{0\}$. For a parameter $w \in \mathcal{W} \subseteq \mathbb{R}^k \setminus \{0\}$, we consider linear decoders of the form:

$$f_w(x) \equiv \operatorname*{arg\,max}_{y \in \mathcal{Y}(x)} \phi(x, y) \cdot w \tag{1}$$

In practice, very few cases of the above general *inference* problem are tractable, while most are NP-hard and also hard to approximate within a fixed factor. (We defer the details in theory of computation to Section 6.)

We also introduce the *distortion* function $d : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$. The value $d(y, y')$ measures the amount of difference between two structured outputs y and y'. Disregarding the computational and statistical aspects, the ultimate goal is to set the parameter w in order to minimize the decoder distortion. That is:

$$\min_{w \in \mathcal{W}} \mathop{\mathbb{E}}_{(x,y) \sim D} [d(y, f_w(x))] \tag{2}$$

Computationally speaking, the above procedure is inefficient since $d(y, f_w(x))$ is a discontinuous function with respect to w and thus, it is in general an exponential-time optimization problem. Statistically speaking, the problem in eq.(2) requires access to the data distribution D and thus, in general it would require an infinite amount of data. In practice, we only have access to a small amount of training data.

Additionally, eq.(2) would potentially favor parameters w with low distortion, but that could be in a neighborhood of parameters with high distortion. In order to avoid this issue, we could optimize a more "robust" objective under Gaussian perturbations. More formally, let $\alpha > 0$ and let $Q(w)$ be a unit-variance Gaussian distribution centered at $w\alpha$ of parameters $w' \in \mathcal{W}$. The Gibbs decoder distortion of the perturbation distribution $Q(w)$ and data distribution D, is defined as:

$$L(Q(w), D) = \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\mathop{\mathbb{E}}_{w'\sim Q(w)}[d(y, f_{w'}(x))]\right] \qquad (3)$$

The minimization of the Gibbs decoder distortion can be expressed as:

$$\min_{w\in\mathcal{W}} L(Q(w), D)$$

The focus of our analysis will be to propose upper bounds of the Gibbs decoder distortion, with good computational and statistical properties. That is, we will propose upper bounds that can be computed in polynomial-time, and that require a small amount of training data.

For our analysis, we follow the same set of assumptions as in (McAllester, 2007). We define the margin $m(x, y, y', w)$ as the amount by which y is preferable to y' under the parameter w. More formally:

$$m(x, y, y', w) \equiv \phi(x, y)\cdot w - \phi(x, y')\cdot w$$

Let $c(p, x, y)$ be a nonnegative integer that gives the number of times that the part $p \in \mathcal{P}$ appears in the pair (x, y). For a part $p \in \mathcal{P}$, we define the feature p as follows:

$$\phi_p(x, y) \equiv c(p, x, y)$$

We let $\mathcal{P}(x) \neq \emptyset$ denote the set of $p \in \mathcal{P}$ such that there exists $y \in \mathcal{Y}(x)$ with $c(p, x, y) > 0$. We define the Hamming distance H as follows:

$$H(x, y, y') \equiv \sum_{p\in\mathcal{P}(x)} |c(p, x, y) - c(p, x, y')|$$

The commonly applied margin-based approach to learning w uses the maximum loss over all possible structured outputs (Altun & Hofmann, 2003; Collins, 2004; Taskar et al., 2003). That is:[1]

$$\min_{w\in\mathcal{W}} \frac{1}{n}\sum_{(x,y)\in S}\max_{\hat{y}\in\mathcal{Y}(x)} d(y,\hat{y})\, 1\begin{pmatrix}H(x,y,\hat{y})\\ -m(x,y,\hat{y},w)\geq 0\end{pmatrix} + \lambda\|w\|_2^2 \qquad (4)$$

In Section 2, we reproduce the results in (McAllester, 2007) and show that the above objective is related to an upper bound of the Gibbs decoder distortion in eq.(3). Note that evaluating the objective function in eq.(4) is as hard as the inference problem in eq.(1), since both perform maximization over the set $\mathcal{Y}(x)$.

Our main contributions are presented in Sections 3 and 4. Inspired by recent work in natural language processing (Zhang et al., 2014; Zhang et al., 2015), we show a tighter upper bound of the Gibbs decoder distortion in eq.(3), which is related to the following objective:[1]

$$\min_{w\in\mathcal{W}} \frac{1}{n}\sum_{(x,y)\in S}\max_{\hat{y}\in T(w,x)} d(y,\hat{y})\, 1\begin{pmatrix}H(x,y,\hat{y})\\ -m(x,y,\hat{y},w)\geq 0\end{pmatrix} + \lambda\|w\|_2^2 \qquad (5)$$

where $T(w, x)$ is a set of random structured outputs sampled i.i.d. from some proposal distribution with support on $\mathcal{Y}(x)$. Note that evaluating the objective function in eq.(5) is linear-time in the number of random structured outputs in $T(w, x)$.

2 FROM PAC-BAYES TO THE MAXIMUM LOSS OVER ALL POSSIBLE STRUCTURED OUTPUTS

In this section, we show the relationship between PAC-Bayes bounds and the commonly used maximum loss over all possible structured outputs.

As reported in (McAllester, 2007), by using the PAC-Bayes framework under Gaussian perturbations, we show that the commonly used maximum loss over all possible structured outputs is an upper bound of the Gibbs decoder distortion up to statistical accuracy ($\mathcal{O}(\sqrt{\log n/n})$ for n training samples).

Theorem 1 (McAllester, 2007)**.** *Assume that there exists a finite integer value ℓ such that $|\cup_{(x,y)\in S}\mathcal{P}(x)| \leq \ell$. Fix $\delta \in (0, 1)$. With probability at least $1 - \delta/2$ over the choice of n training samples, simultaneously for all parameters $w \in \mathcal{W}$ and unit-variance Gaussian perturbation distributions $Q(w)$ centered at $w\sqrt{2\log(2n\ell/\|w\|_2^2)}$, we have:*

$$L(Q(w), D) \leq \frac{1}{n}\sum_{(x,y)\in S}\max_{\hat{y}\in\mathcal{Y}(x)} d(y,\hat{y})\, 1\begin{pmatrix}H(x,y,\hat{y})\\ -m(x,y,\hat{y},w)\geq 0\end{pmatrix} + \frac{\|w\|_2^2}{n} + \sqrt{\frac{\|w\|_2^2\log(2n\ell/\|w\|_2^2) + \log(2n/\delta)}{2(n-1)}}$$

(See Appendix A for detailed proofs.)

The proof of the above is based on the PAC-Bayes theorem and well-known Gaussian concentration inequalities.

[1] For computational convenience, the *convex* hinge loss $\max(0, 1 + z)$ is used in practice instead of the *discontinuous* 0/1 loss $1(z \geq 0)$.

As it is customary in generalization results, a *deterministic* expectation with respect to the data distribution D is upper-bounded by a *stochastic* quantity with respect to the training set S. This takes into account the statistical aspects of the problem.

Note that the upper bound uses maximization with respect to $\mathcal{Y}(x)$ and that in general, $|\mathcal{Y}(x)|$ is exponential with respect to the input size. Thus, the computational aspects of the problem have not been fully addressed yet. In the next section, we solve this issue by introducing randomness.

3 FROM PAC-BAYES TO THE MAXIMUM LOSS OVER RANDOM STRUCTURED OUTPUTS

In this section, we analyze the relationship between PAC-Bayes bounds and the maximum loss over random structured outputs sampled independently from some proposal distribution.

First, we will focus on the computational aspects. Instead of using maximization with respect to $\mathcal{Y}(x)$, we will perform maximization with respect to a set $T(w, x)$ of random structured outputs sampled i.i.d. from some proposal distribution $R(w, x)$ with support on $\mathcal{Y}(x)$. In order for this approach to be computationally appealing, $|T(w, x)|$ should be polynomial, even when $|\mathcal{Y}(x)|$ is exponential with respect to the input size.

Assumptions A and B will allow us to attain $|T(w, x)| = \mathcal{O}\left(\max\left(\frac{1}{\log(1/\beta)}, \|w\|_2^2\right)\right)$. The constant $\beta \in [0, 1)$ is properly introduced on Assumption A. It can be easily observed that β plays an important role in the number of random structured outputs that we need to draw from the proposal distribution $R(w, x)$. Next, we present our first assumption.

Assumption A (Maximal distortion)**.** *The proposal distribution $R(w, x)$ fulfills the following condition. There exists a value $\beta \in [0, 1)$ such that for all $(x, y) \in S$ and $w \in \mathcal{W}$:*

$$\mathop{\mathbb{P}}_{y' \sim R(w,x)}[d(y, y') = 1] \geq 1 - \beta$$

In Section 4 we show examples that fulfill the above assumption, which include a *binary* distortion function for *any* type of structured output, as well as a distortion function that returns the number of different edges/elements for directed spanning trees, directed acyclic graphs and cardinality-constrained sets.

Next, we present our second assumption that allows obtaining $|T(w, x)| = \mathcal{O}\left(\max\left(\frac{1}{\log(1/\beta)}, \|w\|_2^2\right)\right)$. While Assumption A contributes with the term $\frac{1}{\log(1/\beta)}$ in $|T(w, x)|$, the following assumption contributes with the term $\|w\|_2^2$ in $|T(w, x)|$.

Assumption B (Low norm)**.** *For any vector $z \in \mathbb{R}^k$, define:*

$$\mu(z) = \begin{cases} z/\|z\|_1 & \textit{if } z \neq 0 \\ 0 & \textit{if } z = 0 \end{cases}$$

The proposal distribution $R(w, x)$ fulfills the following condition for all $(x, y) \in S$ and $w \in \mathcal{W}$:[2]

$$\left\| \mathop{\mathbb{E}}_{y' \sim R(w,x)}[\mu(\phi(x, y) - \phi(x, y'))] \right\|_2 \leq \frac{1}{2\sqrt{n}} \leq \frac{1}{2\|w\|_2}$$

It is natural to ask whether there are instances that fulfill the above assumption. In Section 4 we provide two extreme cases: one example of a *sparse* mapping and a uniform proposal, and one example of a *dense* mapping and an *arbitrary* proposal distribution.

We will now focus on the statistical aspects. Note that randomness does not only stem from data, but also from sampling structured outputs. That is, in Theorem 1, randomness only stems from the training set S. We now need to produce generalization results that hold for all the sets $T(w, x)$ of random structured outputs. In addition, the uniform convergence of Theorem 1 holds for all parameters w. We now need to produce a generalization result that also holds for all possible proposal distributions $R(w, x)$. Therefore, we need a method for upper-bounding the number of possible proposal distributions $R(w, x)$. Assumption C will allow us to upper-bound this number.

Assumption C (Linearly inducible ordering)**.** *The proposal distribution $R(w, x)$ depends solely on the linear ordering induced by the parameter $w \in \mathcal{W}$ and the mapping $\phi(x, \cdot)$. More formally, let $r(x) \equiv |\mathcal{Y}(x)|$ and thus $\mathcal{Y}(x) \equiv \{y_1 \dots y_{r(x)}\}$. Let $w, w' \in \mathcal{W}$ be any two arbitrary parameters. Let $\pi(x) = (\pi_1 \dots \pi_{r(x)})$ be a permutation of $\{1 \dots r(x)\}$ such that $\phi(x, y_{\pi_1}) \cdot w < \cdots < \phi(x, y_{\pi_{r(x)}}) \cdot w$. Let $\pi'(x) = (\pi'_1 \dots \pi'_{r(x)})$ be a permutation of $\{1 \dots r(x)\}$ such that $\phi(x, y_{\pi'_1}) \cdot w' < \cdots < \phi(x, y_{\pi'_{r(x)}}) \cdot w'$. For all $w, w' \in \mathcal{W}$ and $x \in \mathcal{X}$, if $\pi(x) = \pi'(x)$ then $KL(R(w, x)\|R(w', x)) = 0$. In this case, we say that the proposal distribution fulfills $R(\pi(x), x) \equiv R(w, x)$.*

Assumption C states that two proposal distributions $R(w, x)$ and $R(w', x)$ are the same provided that for the same permutation $\pi(x)$ we have $\phi(x, y_{\pi_1}) \cdot w < \cdots < \phi(x, y_{\pi_{r(x)}}) \cdot w$ and $\phi(x, y_{\pi_1}) \cdot w' < \cdots < \phi(x, y_{\pi_{r(x)}}) \cdot w'$. Geometrically speaking, for a fixed x we first project the feature vectors $\phi(x, y)$ of all the structured outputs $y \in \mathcal{Y}(x)$ onto

[2] The second inequality follows from an implicit assumption made in Theorem 1, i.e., $\|w\|_2^2/n \leq 1$. Note that if $\|w\|_2^2/n > 1$ then Theorem 1 provides an upper bound greater than 1, which is meaningless since the distortion function d is at most 1.

the lines w and w'. Let $\pi(x)$ and $\pi'(x)$ be the resulting ordering of the structured outputs after projecting them onto w and w' respectively. Two proposal distributions $R(w,x)$ and $R(w',x)$ are the same provided that $\pi(x) = \pi'(x)$. That is, the specific values of $\phi(x,y) \cdot w$ and $\phi(x,y) \cdot w'$ are irrelevant, and only their ordering matters.

In Section 4 we show examples that fulfill the above assumption, which include the algorithm proposed in (Zhang et al., 2014; Zhang et al., 2015) for directed spanning trees, and our proposed generalization to any type of data structure with computationally efficient local changes.

In what follows, by using the PAC-Bayes framework under Gaussian perturbations, we show that the maximum loss over random structured outputs sampled independently from some proposal distribution provides an upper bound of the Gibbs decoder distortion up to statistical accuracy ($\mathcal{O}(\log^{3/2} n/\sqrt{n})$ for n training samples).

Theorem 2. *Assume that there exist finite integer values ℓ and r such that $|\cup_{(x,y)\in S} \mathcal{P}(x)| \leq \ell$ and $|\mathcal{Y}(x)| \leq r$ for all $(x,y) \in S$. Assume that the proposal distribution $R(w,x)$ with support on $\mathcal{Y}(x)$ fulfills Assumption A with value β, as well as Assumptions B and C. Fix $\delta \in (0,1)$ and an integer $\mathfrak{s}$ such that $3 \leq \mathfrak{s} \leq \frac{9}{20}\sqrt{\ell+1}$. With probability at least $1-\delta$ over the choice of both n training samples and n sets of random structured outputs, simultaneously for all parameters $w \in \mathcal{W}$ with $\|w\|_0 \leq \mathfrak{s}$, unit-variance Gaussian perturbation distributions $Q(w)$ centered at $w\sqrt{2\log(2n\ell/\|w\|_2^2)}$, and for sets of random structured outputs $T(w,x)$ sampled i.i.d. from the proposal distribution $R(w,x)$ for each training sample $(x,y) \in S$, such that $|T(w,x)| = \left\lceil \frac{1}{2}\max\left(\frac{1}{\log(1/\beta)}, 32\|w\|_2^2\right)\log n \right\rceil$, we have:*

$$
\begin{aligned}
&L(Q(w), D)\\
&\leq \frac{1}{n}\sum_{(x,y)\in S} \max_{\hat{y}\in T(w,x)} d(y,\hat{y})\, 1\begin{pmatrix} H(x,y,\hat{y}) \\ -m(x,y,\hat{y},w) \geq 0 \end{pmatrix}\\
&+ \frac{\|w\|_2^2}{n} + \sqrt{\frac{\|w\|_2^2 \log(2n\ell/\|w\|_2^2) + \log(2n/\delta)}{2(n-1)}} + \sqrt{\frac{1}{n}}\\
&+ \max\Big(\tfrac{1}{\log(1/\beta)}, 32\|w\|_2^2\Big)\sqrt{\frac{\mathfrak{s}\log(\ell+1)\log^3(n+1)}{n}}\\
&+ 3\sqrt{\frac{\mathfrak{s}(\log\ell + 2\log(nr)) + \log(4/\delta)}{n}}
\end{aligned}
$$

(See Appendix A for detailed proofs.)

The proof of the above is based on Theorem 1 as a starting point. In order to account for the computational aspect of requiring sets $T(w,x)$ of polynomial size, we use Assumptions A and B for bounding a *deterministic* expectation. In order to account for the statistical aspects, we use Assumption C and Rademacher complexity arguments for bounding a *stochastic* quantity for all sets $T(w,x)$ of random structured outputs and all possible proposal distributions $R(w,x)$. The assumption of sparsity (i.e., $\|w\|_0 \leq \mathfrak{s}$) is pivotal for obtaining terms of order $\mathcal{O}(\sqrt{\mathfrak{s}\log \ell/n})$. Without sparsity, the terms would be of order $\mathcal{O}(\sqrt{\ell/n})$ which is not suited for high-dimensional settings.

3.1 Inference on Test Data

Note that the upper bound in Theorem 2 holds simultaneously for all parameters $w \in \mathcal{W}$. Therefore, our result implies that after learning the optimal parameter $\widehat{w} \in \mathcal{W}$ in eq.(5) from *training* data, we can bound the decoder distortion when performing *exact* inference on *test* data. More formally, Theorem 2 can be additionally invoked for a *test* set S', also with probability at least $1-\delta$. Thus, under the same setting as of Theorem 2, the Gibbs decoder distortion is upper-bounded with probability at least $1-2\delta$ over the choice of S and S'. In this paper, we focus on learning the parameter of structured prediction models. We leave the analysis of *approximate* inference on test data for future work.

4 EXAMPLES

In this section, we provide several examples that fulfill the three main assumptions of our theoretical result.

4.1 Examples for the Maximal Distortion Assumption

In what follows, we present some examples that fulfill our Assumption A. For a *binary* distortion function, we show that *any* type of structured output fulfills the above assumption. For a distortion function that returns the number of different edges/elements, we show that directed spanning trees, directed acyclic graphs and cardinality-constrained sets, fulfill the assumption as well.

For simplicity of analysis, most proofs in this part will assume a uniform proposal distribution $R(w,x) = R(x)$ with support on $\mathcal{Y}(x)$. In the following claim, we argue that we can perform a change of measure between different proposal distributions. Thus, allowing us to focus on uniform proposals afterwards.

Claim i (Change of measure)**.** *Let $R(w,x)$ and $R'(w,x)$ two proposal distributions, both with support on $\mathcal{Y}(x)$. Assume that the proposal distribution $R(w,x)$ fulfills Assumption A with value β_1. Let $r_{w,x}(\cdot)$ and $r'_{w,x}(\cdot)$ be the probability mass functions of $R(w,x)$ and $R'(w,x)$ respectively. Assume that the total variation distance between $R(w,x)$ and $R'(w,x)$ is bounded as follows for all $(x,y) \in S$ and $w \in \mathcal{W}$:*

$$
\begin{aligned}
TV(R(w,x)\|R'(w,x)) &\equiv \frac{1}{2}\sum_{y\in\mathcal{Y}(x)} |r_{w,x}(y) - r'_{w,x}(y)|\\
&\leq \beta_2
\end{aligned}
$$

The proposal distribution $R'(w,x)$ fulfills Assumption A with $\beta = \beta_1 + \beta_2$ provided that $\beta_1 + \beta_2 \in [0,1)$.

Next, we provide a result for *any* type of structured output, but for a *binary* distortion function.

Claim ii (Any type of structured output)**.** *Let $\mathcal{Y}(x)$ be an arbitrary countable set of feasible decodings of x, such that $|\mathcal{Y}(x)| \geq 2$ for all $(x,y) \in S$. Let $d(y,y') = 1(y \neq y')$. The uniform proposal distribution $R(w,x) = R(x)$ with support on $\mathcal{Y}(x)$ fulfills Assumption A with $\beta = 1/2$.*

The following claim pertains to directed spanning trees and for a distortion function that returns the number of different edges.

Claim iii (Directed spanning trees)**.** *Let $\mathcal{Y}(x)$ be the set of directed spanning trees of v nodes. Let $A(y)$ be the adjacency matrix of $y \in \mathcal{Y}(x)$. Let $d(y,y') = \frac{1}{2(v-1)} \sum_{ij} |A(y)_{ij} - A(y')_{ij}|$. The uniform proposal distribution $R(w,x) = R(x)$ with support on $\mathcal{Y}(x)$ fulfills Assumption A with $\beta = \frac{v-2}{v-1}$.*

The next result is for directed acyclic graphs and for a distortion function that returns the number of different edges.

Claim iv (Directed acyclic graphs)**.** *Let $\mathcal{Y}(x)$ be the set of directed acyclic graphs of v nodes and b parents per node, such that $2 \leq b \leq v-2$. Let $A(y)$ be the adjacency matrix of $y \in \mathcal{Y}(x)$. Let $d(y,y') = \frac{1}{b(2v-b-1)} \sum_{ij} |A(y)_{ij} - A(y')_{ij}|$. The uniform proposal distribution $R(w,x) = R(x)$ with support on $\mathcal{Y}(x)$ fulfills Assumption A with $\beta = \frac{b^2+2b+2}{b^2+3b+2}$.*

The final example is for cardinality-constrained sets and for a distortion function that returns the number of different elements.

Claim v (Cardinality-constrained sets)**.** *Let $\mathcal{Y}(x)$ be the set of sets of b elements chosen from v possible elements, such that $b \leq v/2$. Let $d(y,y') = \frac{1}{2b}(|y - y'| + |y' - y|)$. The uniform proposal distribution $R(w,x) = R(x)$ with support on $\mathcal{Y}(x)$ fulfills Assumption A with $\beta = 1/2$.*

4.2 Examples for the Low Norm Assumption

Next, we present some examples that fulfill our Assumption B. We provide two extreme cases: one example for *sparse* mappings, and one example for *dense* mappings.

Next, we provide a result for a particular instance of a sparse mapping and a uniform proposal distribution.

Claim vi (Sparse mapping)**.** *Let $b > 0$ be an arbitrary integer value. For all $(x,y) \in S$, let $\mathcal{Y}(x) = \cup_{p \in \mathcal{P}(x)} \mathcal{Y}_p(x)$, where the partition $\mathcal{Y}_p(x)$ is defined as follows:*

$$(\forall p \in \mathcal{P}(x))\ \mathcal{Y}_p(x) \equiv \{y' \mid |\phi_p(x,y) - \phi_p(x,y')| = b \wedge (\forall q \neq p)\ \phi_q(x,y) = \phi_q(x,y')\}$$

If $n \leq |\mathcal{P}(x)|/4$ for all $(x,y) \in S$, then the uniform proposal distribution $R(w,x) = R(x)$ with support on $\mathcal{Y}(x)$ fulfills Assumption B.

The following claim pertains to a particular instance of a dense mapping and an *arbitrary* proposal distribution.

Claim vii (Dense mapping)**.** *Let $b > 0$ be an arbitrary integer value. Let $|\phi_p(x,y) - \phi_p(x,y')| = b$ for all $(x,y) \in S$, $y' \in \mathcal{Y}(x)$ and $p \in \mathcal{P}(x)$. If $n \leq |\mathcal{P}(x)|/4$ for all $(x,y) \in S$, then any arbitrary proposal distribution $R(w,x)$ fulfills Assumption B.*

4.3 Examples for the Linearly Inducible Ordering Assumption

In what follows, we present some examples that fulfill our Assumption C. We show that the algorithm proposed in (Zhang et al., 2014; Zhang et al., 2015) for directed spanning trees, fulfills the above assumption. We also generalize the algorithm in (Zhang et al., 2014; Zhang et al., 2015) to any type of data structure with computationally efficient local changes, and show that this generalization fulfills the assumption as well.

Next, we present the algorithm proposed in (Zhang et al., 2014; Zhang et al., 2015) for dependency parsing in natural language processing. Here, x is a sentence of v words and $\mathcal{Y}(x)$ is the set of directed spanning trees of v nodes.

Algorithm 1 Procedure for sampling a directed spanning tree $y' \in \mathcal{Y}(x)$ from a greedy local proposal distribution $R(w,x)$

```
Input: parameter $w \in \mathcal{W}$, sentence $x \in \mathcal{X}$
Draw uniformly at random a directed spanning tree $\hat{y} \in \mathcal{Y}(x)$
repeat
  $s \leftarrow$ post-order traversal of $\hat{y}$
  for each node $t$ in the list $s$ do
    for each node $u$ before $t$ in the list $s$ do
      $y \leftarrow$ change the parent of node $t$ to $u$ in $\hat{y}$
      if $\phi(x,y) \cdot w > \phi(x,\hat{y}) \cdot w$ then
        $\hat{y} \leftarrow y$
      end if
    end for
  end for
until no refinement in last iteration
Output: directed spanning tree $y' \leftarrow \hat{y}$
```

The above algorithm has the following property:

Claim viii (Sampling for directed spanning trees)**.** *Algorithm 1 fulfills Assumption C.*

Note that Algorithm 1 proposed in (Zhang et al., 2014; Zhang et al., 2015) uses the fact that we can perform local changes to a directed spanning tree in a computationally efficient manner. That is, changing parents of nodes in a post-order traversal will produce directed spanning trees. We can extend the above algorithm to any type of data structure where we can perform computationally efficient local changes. For instance, we can easily extend the method for directed acyclic graphs (traversed in post-order as well) and for sets with up to some prespecified number of elements.

Next, we generalize Algorithm 1 to any type of structured output.

Algorithm 2 Procedure for sampling a structured output $y' \in \mathcal{Y}(x)$ from a greedy local proposal distribution $R(w, x)$

```
Input: parameter $w \in \mathcal{W}$, observed input $x \in \mathcal{X}$
Draw uniformly at random a structured output $\hat{y} \in \mathcal{Y}(x)$
repeat
    Make a local change to $\hat{y}$ in order to increase $\phi(x, \hat{y}) \cdot w$
until no refinement in last iteration
Output: structured output $y' \leftarrow \hat{y}$
```

The above algorithm has the following property:

Claim ix (Sampling for any type of structured output)**.** *Algorithm 2 fulfills Assumption C.*

5 EXPERIMENTAL RESULTS

In this section, we provide experimental evidence on synthetic data. Note that the work of (Zhang et al., 2014; Zhang et al., 2015) has provided extensive experimental evidence on real-world datasets, for part-of-speech tagging and dependency parsing in the context of natural language processing. Our experimental results are not only for directed spanning trees (Zhang et al., 2014; Zhang et al., 2015) but also for directed acyclic graphs and cardinality-constrained sets.

We performed 30 repetitions of the following procedure. We generated a ground truth parameter w^* with independent zero-mean and unit-variance Gaussian entries. Then, we generated a training set S of $n = 100$ samples. The fixed mapping ϕ from pairs (x, y) to feature vectors $\phi(x, y)$ is as follows. For every pair of possible edges/elements i and j, we define $\phi_{ij}(x, y) = 1\,(x_{ij} = 1 \wedge i \in y \wedge j \in y)$. For instance, for directed spanning trees of v nodes, we have $x \in \{0, 1\}^{\binom{v}{2}}$ and $\phi(x, y) \in \mathbb{R}^{\binom{v}{2}}$. In order to generate each training sample $(x, y) \in S$, we generated a random vector x with independent Bernoulli entries, each with equal probability of being 1 or 0. After generating x, we set $y = f_{w^*}(x)$. That is, we solved eq.(1) in order to produce the latent structured output y from the observed input x and the parameter w^*.

We compared two training methods: the maximum loss over all possible structured outputs as in eq.(4), and the maximum loss over random structured outputs as in eq.(5). For both minimization problems, we replaced the *discontinuous* 0/1 loss $1\,(z \geq 0)$ with the *convex* hinge loss $\max\,(0, 1 + z)$, as it is customary. For both problems, we used $\lambda = 1/n$ as suggested by Theorems 1 and 2, and we performed 20 iterations of the subgradient descent method with a decaying step size $1/\sqrt{t}$ for iteration t. For sampling random structured outputs in eq.(5), we implemented Algorithm 2 for directed spanning trees, directed acyclic graphs and cardinality-constrained sets. 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. We used $\beta = 0.8$ for directed spanning trees, $\beta = 0.85$ for directed acyclic graphs, and $\beta = 0.5$ for cardinality-constrained sets, as prescribed by Claims iii, iv and v. After training, for inference on an independent test set, we used eq.(1) for the maximum loss over all possible structured outputs. For the maximum loss over random structured outputs, we use the following *approximate* inference approach:

$$\widetilde{f}_w(x) \equiv \operatorname*{arg\,max}_{y \in T(w,x)} \phi(x, y) \cdot w \qquad (6)$$

Table 1 shows the average over 30 repetitions, and the standard error at 95% confidence level of the following measurements. We report the runtime, the training distortion as well as the test distortion in an independently generated set of 100 samples. We also report the normalized distance of the learnt $\widehat{w}$ to the ground truth w^*, i.e., $\|\widehat{w} - w^*\|_2/\sqrt{\ell}$. Additionally, we report the angle of the learnt $\widehat{w}$ with respect to the ground truth w^*, i.e. $\arccos(\widehat{w} \cdot w^*/(\|\widehat{w}\|_2\|w^*\|_2))$. In the different study cases (directed spanning trees, directed acyclic graphs and cardinality-constrained sets), the maximum loss over random structured outputs outperforms the maximum loss over all possible structured outputs.

6 DISCUSSION

In this section, we provide more details regarding the computational complexity of the inference problem. We also present a brief review of the previous work and provide ideas for extending our theoretical result.

6.1 Computational Complexity of the Inference Problem

Very few cases of the general *inference* problem in eq.(1) are tractable. For instance, if $\mathcal{Y}(x)$ is the set of directed spanning trees, and w is a vector of edge weights (i.e., linear with respect to y), then eq.(1) is equivalent to the maximum directed spanning tree problem, which is polynomial-time. In general, the inference problem in eq.(1) is not

Table 1: Average over 30 repetitions, and standard error at 95% confidence level of several methods and measurements. For the maximum loss over all possible structured outputs (All) we used eq.(4) for training, and eq.(1) for inference on a test set. For the maximum loss over random structured outputs (Random and Random/All) we used eq.(5) for training. For inference, Random used eq.(6) while Random/All used eq.(1). Random outperforms All in the different study cases (directed spanning trees, directed acyclic graphs and cardinality-constrained sets). The difference between Random and Random/All is not statistically significant.

Problem	**Method**	**Training runtime**	**Training distortion**	**Test runtime**	**Test distortion**	**Distance to ground truth**	**Angle with ground truth**
Directed spanning trees	All	1000	$52\% \pm 1.1\%$	12.4 ± 0.4	$61\% \pm 1.8\%$	0.56 ± 0.004	$74° \pm 0.3°$
	Random	104 ± 3	$38\% \pm 2.1\%$	2.4 ± 0.1	$56\% \pm 1.9\%$	0.51 ± 0.005	$49° \pm 0.6°$
	Random/All			12.4 ± 0.3	$56\% \pm 1.9\%$		
Directed acyclic graphs	All	1000	$41\% \pm 1.2\%$	10.8 ± 0.2	$45\% \pm 1.5\%$	0.60 ± 0.020	$61° \pm 1.0°$
	Random	386 ± 21	$30\% \pm 1.3\%$	8.5 ± 0.2	$39\% \pm 1.6\%$	0.40 ± 0.008	$37° \pm 1.0°$
	Random/All			10.8 ± 0.2	$39\% \pm 1.6\%$		
Cardinality constrained sets	All	1000	$42\% \pm 1.4\%$	11.1 ± 0.4	$45\% \pm 1.8\%$	0.58 ± 0.011	$65° \pm 0.6°$
	Random	272 ± 9	$21\% \pm 1.2\%$	6.0 ± 0.2	$30\% \pm 1.9\%$	0.44 ± 0.008	$30° \pm 0.8°$
	Random/All			10.9 ± 0.3	$29\% \pm 2.1\%$		

only NP-hard but also hard to approximate. For instance, if $\mathcal{Y}(x)$ is the set of directed acyclic graphs, and w is a vector of edge weights (i.e., linear with respect to y), then eq.(1) is equivalent to the maximum acyclic subgraph problem, which approximating within a factor better than $1/2$ is unique-games hard (Guruswami et al., 2008). As an additional example, consider the case where $\mathcal{Y}(x)$ is the set of sets with up to some prespecified number of elements (i.e., $\mathcal{Y}(x)$ is a cardinality constraint), and the objective $\phi(x, y) \cdot w$ is submodular with respect to y. In this case, eq.(1) cannot be approximated within a factor better than $1 - 1/e$ unless P=NP (Nemhauser et al., 1978).

These negative results made us to avoid interpreting the maximum loss over random structured outputs in eq.(5) as an approximate optimization algorithm for the maximum loss over all possible structured outputs in eq.(4).

6.2 Previous Work

Approximate inference was proposed in (Kulesza & Pereira, 2007), with an adaptation of the proof techniques in (McAllester, 2007). More specifically, (Kulesza & Pereira, 2007) performs maximization of the loss over a *superset* of feasible decodings of x, i.e., over $y \in \mathcal{Y}'(x) \supseteq \mathcal{Y}(x)$. Note that our upper bound of the Gibbs decoder distortion dominates the maximum loss over $y \in \mathcal{Y}(x)$, and the latter dominates the upper bound of (Kulesza & Pereira, 2007). One could potentially use a similar argument with respect to a *subset* of feasible decodings of x, i.e., with respect to $y \in \mathcal{Y}'(x) \subseteq \mathcal{Y}(x)$. Unfortunately, this approach does not obtain an upper bound of the Gibbs decoder distortion.

Tangential to our work, previous analyses have exclusively focused either on sample complexity or convergence. Sample complexity analyses include margin bounds (Taskar et al., 2003), Rademacher complexity (London et al., 2013) and PAC-Bayes bounds (McAllester, 2007; McAllester & Keshet, 2011). Convergence have been analyzed for specific algorithms for the separable (Collins & Roark, 2004) and nonseparable (Crammer et al., 2006) cases.

6.3 Concluding Remarks

The work of (Zhang et al., 2014; Zhang et al., 2015) has shown extensive experimental evidence for part-of-speech tagging and dependency parsing in the context of natural language processing. In this paper, we present a theoretical analysis that explains the experimental success of (Zhang et al., 2014; Zhang et al., 2015) for directed spanning trees. Our analysis was provided for a far more general setup, which allowed proposing algorithms for other types of structured outputs, such as directed acyclic graphs and cardinality-constrained sets. We hope that our theoretical work will motivate experimental validation on many other real-world structured prediction problems.

There are several ways of extending this research. While we focused on Gaussian perturbations, it would be interesting to analyze other distributions from the computational as well as statistical viewpoints. We analyzed a general class of proposal distributions that depend on the induced linear orderings. Algorithms that make greedy local changes, traverse the set of feasible decodings in a constrained fashion, by following allowed moves defined by some prespecified graph. The addition of these graph-theoretical constraints would enable obtaining tighter upper bounds. From a broader perspective, extensions of our work to latent models (Ping et al., 2014; Yu & Joachims, 2009) as well as maximum a-posteriori perturbation models (Gane et al., 2014; Papandreou & Yuille, 2011) would be of great interest. Finally, while we focused on learning the parameter of structured prediction models, it would be interesting to analyze *approximate* inference for prediction on an independent test set.

References

Altun, Y., & Hofmann, T. (2003). Large margin methods for label sequence learning. *European Conference on Speech Communication and Technology*, 145–152.

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

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

Collins, M. (2004). Parameter estimation for statistical parsing models: Theory and practice of distribution-free methods. In *New developments in parsing technology*, vol. 23, 19–55. Kluwer Academic.

Collins, M., & Roark, B. (2004). Incremental parsing with the perceptron algorithm. *Annual Meeting of the Association for Computational Linguistics*, 111–118.

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

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., & Singer, Y. (2006). Online passive-aggresive algorithms. *Journal of Machine Learning Research*, *7*, 551–585.

Gane, A., Hazan, T., & Jaakkola, T. (2014). Learning with maximum a-posteriori perturbation models. *International Conference on Artificial Intelligence and Statistics*, *33*, 247–256.

Guruswami, V., Manokaran, R., & Raghavendra, P. (2008). Beating the random ordering is hard: Inapproximability of maximum acyclic subgraph. *Foundations of Computer Science*, 573–582.

Kulesza, A., & Pereira, F. (2007). Structured learning with approximate inference. *Neural Information Processing Systems*, *20*, 785–792.

London, B., Huang, B., Taskar, B., & Getoor, L. (2013). Collective stability in structured prediction: Generalization from one example. *International Conference on Machine Learning*, 828–836.

McAllester, D. (2007). Generalization bounds and consistency. In *Predicting structured data*, 247–261. MIT Press.

McAllester, D., & Keshet, J. (2011). Generalization bounds and consistency for latent structural probit and ramp loss. *Neural Information Processing Systems*, *24*, 2205–2212.

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

Neylon, T. (2006). *Sparse solutions for linear prediction problems*. Doctoral dissertation, New York University.

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

Ping, W., Liu, Q., & Ihler, A. (2014). Marginal structured SVM with hidden variables. *International Conference on Machine Learning*, 190–198.

Taskar, B., Guestrin, C., & Koller, D. (2003). Max-margin Markov networks. *Neural Information Processing Systems*, *16*, 25–32.

Yu, C., & Joachims, T. (2009). Learning structural SVMs with latent variables. *International Conference on Machine Learning*, 1169–1176.

Zhang, Y., Lei, T., Barzilay, R., & Jaakkola, T. (2014). Greed is good if randomized: New inference for dependency parsing. *Empirical Methods in Natural Language Processing*, 1013–1024.

Zhang, Y., Li, C., Barzilay, R., & Darwish, K. (2015). Randomized greedy inference for joint segmentation, POS tagging and dependency parsing. *North American Chapter of the Association for Computational Linguistics*, 42–52.