Polyhedral Outer Approximations with Application to Natural Language Parsing

André F. T. Martins[†‡] AFM@CS.CMU.EDU
Noah A. Smith[†] NASMITH@CS.CMU.EDU
Eric P. Xing[†] EPXING@CS.CMU.EDU
[†]School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA
[‡]Instituto de Telecomunicações, Instituto Superior Técnico, Lisboa, Portugal

Abstract

Recent approaches to learning structured predictors often require approximate inference for tractability; yet its effects on the learned model are unclear. Meanwhile, most learning algorithms act as if computational cost was constant within the model class. This paper sheds some light on the first issue by establishing risk bounds for max-margin learning with LP relaxed inference and addresses the second issue by proposing a new paradigm that attempts to penalize "time-consuming" hypotheses. Our analysis relies on a geometric characterization of the outer polyhedra associated with the LP relaxation. We then apply these techniques to the problem of dependency parsing, for which a concise LP formulation is provided that handles non-local output features. A significant improvement is shown over arc-factored models.

1. Introduction

Structured classification tackles problems characterized by strong interdependence among the output variables, often with a sequential, graphical, or combinatorial structure. Problems of this kind arise in natural language processing (NLP), computer vision, robotics, and computational biology. Considerable progress has been made lately toward a unified view of these problems (Lafferty et al., 2001; Collins, 2002; Taskar et al., 2004a; Tsochantaridis et al., 2004).

A typical approach is to capture the problem structure via a Markov network. Unfortunately, exact inference and learning are only tractable for a small class of network topologies. While discriminative learning is able to handle features that depend globally on the *input* variables, the analogous property on the *output* side is more difficult to obtain. For tractability, assumptions about the *locality* of output variable dependence are often made at the expense of expressive power. Exploring non-local output dependencies is not just an empiricist attempt to produce more accurate models: indeed, there are instances of structured problems in which the output variables are *known* to interact globally, e.g., by definition of the output space. An example is natural language parsing, where the output variables must "agree" to ensure that they jointly encode a valid parse tree. While dynamic programming sometimes offers a solution to this problem (albeit under locality assumptions), the same does not happen in many combinatorial problems of interest, like those involving matchings, permutations, or spanning trees.

When exact inference is intractable, one has to resort to approximate algorithms; typically, the same algorithms are called as subroutines to train the model. This has been done, e.g., by Taskar et al. (2004b) and Daumé and Marcu (2005); recently, Kulesza and Pereira (2007) and Finley and Joachims (2008) provided some learning approximation guarantees and empirical analyses. However, a theoretical study of the actual impact of these approximations in the learning procedure—when compared with the exact formulation—is still missing.

This paper aims to fill this gap for the case of *outer* approximations when learning large margin classifiers. Often, the problem of inference can be represented as an integer linear program (ILP); this is common in NLP applications like semantic role labeling (Roth & Yih, 2005), summarization (Clarke & Lapata, 2008), coreference resolution (Denis & Baldridge, 2007), and dependency parsing (Riedel & Clarke, 2006; Martins

Appearing in *Proceedings of the 26th International Conference on Machine Learning*, Montreal, Canada, 2009.

et al., 2009). While solving an ILP is NP-hard in general, fast solvers are available today that make this a practical solution in some cases. We study approximate techniques based on LP relaxations applied to the problem of dependency parsing; our formulation effectively handles global features and constraints. The techniques discussed here are also relevant to other applications; in particular, they are able to exploit expert knowledge in the form of soft or hard first-order constraints (Richardson & Domingos, 2006; Chang et al., 2008).

The contributions of this paper are:

- We provide risk bounds for approximate learners, in a large margin framework, that arise from an LP relaxation of the inference problem. We characterize these bounds in terms of geometric and algorithmic properties. In particular, we provide sufficient conditions for algorithmic separability.
- We propose a new learning paradigm for approximate inference that balances accuracy and runtime, where an additional loss term is included in the learner objective function to penalize models with long expected runtime. We formulate and analyze a new algorithm based on this paradigm.
- We empirically evaluate the performance of such approximate learners on dependency parsing tasks. Our parsers are able to handle sibling and grandparent interactions, word valency, and to softly favor nearly projective parses.

The paper is organized as follows. Sec. 2 introduces the framework of ILP formulations for large margin structured classification; Sec. 3 provides a theoretical analysis of outer approximations through LP relaxations; Sec. 4 presents the task; experiments are discussed in Sec. 5. We conclude in Sec. 6.

2. Structured Classification and LP

Let $\mathcal{X}$ and $\mathcal{Y}$ denote the input and output sets, respectively, and assume a supervised setting, where we are given labeled data $\mathcal{L} \triangleq \{(x_1, y_1), \ldots, (x_m, y_m)\} \subseteq \mathcal{X} \times \mathcal{Y}$, drawn according to a fixed, unknown distribution $P(X, Y)$, and aim to learn a function $h : \mathcal{X} \to \mathcal{Y}$ with small expected loss $\mathbb{E}\ell(h(X); Y)$ on unseen data; here, $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ denotes the loss function.

We are interested in the case where $\mathcal{Y}$ is exponentially large but finite. We assume that there is a bijection ζ between $\mathcal{Y}$ and the set of vertices of a polytope $\mathcal{Z} \triangleq \{\mathbf{z} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{z} \leq \mathbf{b}\}$, where $\mathbf{A}$ is a p-by-n matrix and $\mathbf{b}$ is a vector in $\mathbb{R}^p$, for some integers p and n. In that case, to carry out structured classification (the **inference** problem), one may transform the problem of optimizing over $\mathcal{Y}$ into that of optimizing a linear function over $\mathcal{Z}$ (guaranteed to attain the optimum at a vertex of $\mathcal{Z}$) and then invert, $y^* = \zeta^{-1}(\mathbf{z}^*)$. This framework was first studied by Taskar et al. (2004b) in the context of associative Markov networks.

Denote by $V(\mathcal{Z})$ the set of vertices of $\mathcal{Z}$. A map $\zeta : \mathcal{Y} \to V(\mathcal{Z})$ arises naturally whenever the elements of $\mathcal{Y}$ can be represented as collections of "parts" in a finite set $\mathcal{R}$ (i.e., each $y \in \mathcal{Y}$ satisfies $y \subseteq \mathcal{R}$).[1] Indeed, let us consider linear classifiers of the form $h_{\mathbf{w}}(x) = \arg\max_{y \in \mathcal{Y}} \mathbf{w}^\top \mathbf{f}(x, y)$ (here, $\mathbf{f}(x, y)$ is a vector of features and $\mathbf{w}$ is the corresponding weight vector). If the features decompose over the parts through

$$\mathbf{f}(x, y) \triangleq \sum_{r \in y} \mathbf{f}_r(x) = \sum_{r \in \mathcal{R}} z_r \mathbf{f}_r(x), \tag{1}$$

where $z_r \triangleq \mathbb{I}(r \in y)$, then, by defining the indicator vector $\mathbf{z} \triangleq (z_r)_{r \in \mathcal{R}}$ and the matrix $\mathbf{F} \triangleq (\mathbf{f}_r(x))_{r \in \mathcal{R}}$:

$$h_{\mathbf{w}}(x) = \arg\max_{y \in \mathcal{Y}} \mathbf{w}^\top \mathbf{f}(x, y) = \zeta^{-1}\left(\arg\max_{\mathbf{z} \in \mathcal{Z}} \mathbf{w}^\top \mathbf{F}\mathbf{z}\right), \tag{2}$$

where $\mathcal{Z}$ is the convex hull of the set of indicator vectors that correspond to elements of $\mathcal{Y}$. Therefore, in this situation, inference can be cast as an LP; this automatically enables learning using any algorithm that only needs to perform inference steps, like the structured perceptron (Collins, 2002). Furthermore, whenever the loss function can be expressed as a linear function of $\mathbf{z}$—which turns out to be the case in many cases of interest, e.g. in Hamming-like losses,

$$\begin{aligned} \ell(y'; y) &\triangleq \sum_{r \in \mathcal{R}} (\mathbb{I}(r \in y')\mathbb{I}(r \notin y) + \mathbb{I}(r \notin y')\mathbb{I}(r \in y)) \\ &= \sum_{r \in \mathcal{R}} z'_r(1 - z_r) + (1 - z'_r)z_r \\ &= \mathbf{p}^\top \mathbf{z}' + q, \end{aligned} \tag{3}$$

where $\mathbf{p} \triangleq \mathbf{1} - 2\mathbf{z}$, $q \triangleq \mathbf{1}^\top \mathbf{z}$, and we made the variable changes $\mathbf{z} \triangleq \zeta(y)$ and $\mathbf{z}' \triangleq \zeta(y')$—then this setting also allows learning the model parameters $\mathbf{w}$ using a max-margin criterion. To see how, define, for each data point $(x_t, y_t) \in \mathcal{L}$, the non-negative quantity

$$\begin{aligned} r_t(\mathbf{w}) &\triangleq \max_{y'_t \in \mathcal{Y}} \mathbf{w}^\top \mathbf{f}(x_t, y'_t) - \mathbf{w}^\top \mathbf{f}(x_t, y_t) + \ell(y'_t; y_t) \\ &= \left(\max_{\mathbf{z}'_t \in \mathcal{Z}} (\mathbf{F}_t^\top \mathbf{w} + \mathbf{p}_t)^\top \mathbf{z}'_t\right) - (\mathbf{F}_t^\top \mathbf{w})^\top \mathbf{z}_t + q_t, \end{aligned} \tag{4}$$

where $\mathbf{z}_t \triangleq \zeta(y_t)$; the problem of finding the arg max in (4) is often referred to as the **loss-augmented inference** (LAI) problem; observe that it can also be

[1] The set of parts $\mathcal{R}$ can be, e.g., the set of all possible clique assignments in a Markov network.

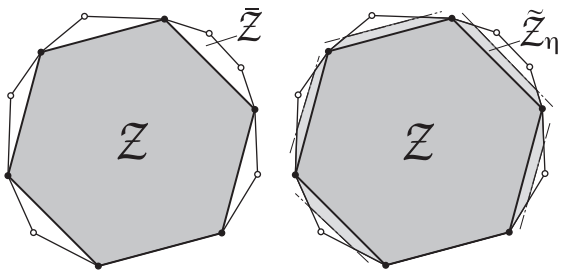

Figure 1. Left: Schematic representation of the outer polytope $\bar{\mathcal{Z}}$ associated with the relaxation (6). Right: The carved polytope $\tilde{\mathcal{Z}}_\eta$ implicit in the problem (13).

cast as an LP. The complete learning problem is

$$\min_{\mathbf{w}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{m}\sum_{t=1}^{m} r_t(\mathbf{w}), \tag{5}$$

where $\lambda \geq 0$ is a regularization parameter. This is a convex problem for which several algorithms have been proposed (Taskar et al., 2004b; Taskar et al., 2006; Tsochantaridis et al., 2004; Ratliff et al., 2006).

In what follows, we denote $\mathbb{U} \triangleq [0,1]$, and $\mathbb{B} \triangleq \{0,1\} = \mathbb{U} \cap \mathbb{Z}$. Although in some special cases there exists a concise polyhedral representation of $\mathcal{Z}$ (in terms of the matrix $\mathbf{A}$ and vector $\mathbf{b}$) that does not happen in general. Often, what we have is a concise representation of an **outer polytope** $\bar{\mathcal{Z}} \supseteq \mathcal{Z}$ such that

$$\min_{\mathbf{z}\in\mathcal{Z}} \mathbf{c}^\top\mathbf{z} = \min_{\mathbf{z}\in\bar{\mathcal{Z}},\mathbf{z}\in\mathbb{B}^n} \mathbf{c}^\top\mathbf{z} \;\geq\; \min_{\mathbf{z}\in\bar{\mathcal{Z}}} \mathbf{c}^\top\mathbf{z} \tag{6}$$

holds for any $\mathbf{c}$ with a fairly tight bound. Notice that, if we assume that $\bar{\mathcal{Z}} \subseteq \mathbb{U}^n$, then there are no integer points in the relative interior of $\bar{\mathcal{Z}}$; consequently, (6) implies that any vertex of $\mathcal{Z}$ is also a vertex of $\bar{\mathcal{Z}}$. The two polytopes are represented schematically in Fig. 1.

3. Learning with LP Relaxed Inference

We now study the impact of relaxations like (6) in the learning problem.

3.1. Approximation Bounds

To cope with our approximate learning setting, we identify $\mathcal{Y} \simeq V(\mathcal{Z})$ through the map ζ; with some abuse of notation, we write $\ell(\mathbf{z}';\mathbf{z})$ for the loss function (3) instead of $\ell(\zeta^{-1}(\mathbf{z}');\zeta^{-1}(\mathbf{z}))$. Also, we extend its domain by defining $\ell(\bar{\mathbf{z}}';\mathbf{z}) = \mathbf{p}^\top\bar{\mathbf{z}}' + q$ for any $\bar{\mathbf{z}}' \in \bar{\mathcal{Z}}$. Observe that, whenever $\mathbf{z} \in V(\mathcal{Z})$, $\ell(\bar{\mathbf{z}}';\mathbf{z}) = \|\bar{\mathbf{z}}' - \mathbf{z}\|_1$. As a consequence, ℓ has the triangle inequality property, i.e., $\ell(\bar{\mathbf{z}}';\mathbf{z}) \leq \ell(\bar{\mathbf{z}}';\mathbf{z}') + \ell(\mathbf{z}';\mathbf{z})$, for any $\bar{\mathbf{z}}' \in \bar{\mathcal{Z}}$ and $\mathbf{z},\mathbf{z}' \in V(\mathcal{Z})$. This fact will be exploited below.

Let $H \triangleq \{h_{\mathbf{w}} \mid \mathbf{w} \in \mathcal{W}\}$ be our hypothesis class, where $\mathcal{W} \subseteq \mathbb{R}^d$ is a convex set, and $h_{\mathbf{w}}$ is the maximum (over $\mathcal{Z}$) of linear discriminant functions, as in (2). We consider approximate inference algorithms $\mathcal{A}$, that accept as input a parameter $\mathbf{w} \in \mathcal{W}$ and a data point $x \in \mathcal{X}$, and output a value $\mathcal{A}(x;\mathbf{w})$ in $\bar{\mathcal{Z}}$. Following Kulesza and Pereira (2007), we say that the dataset $\mathcal{L}$ is **separable** (w.r.t. H) if there is $\mathbf{w} \in \mathcal{W}$ such that $\sum_{t=1}^m \ell(h_{\mathbf{w}}(x_t);\mathbf{z}_t) = 0$; we say that $\mathcal{L}$ is **algorithmically separable** (w.r.t. $\mathcal{A}$) if there is $\mathbf{w} \in \mathcal{W}$ such that $\sum_{t=1}^m \ell(\mathcal{A}(x_t;\mathbf{w});\mathbf{z}_t) = 0$. In both cases, we say that "$\mathcal{L}$ is separated (resp. algorithmically separated) by $\mathbf{w}$" when $\mathbf{w}$ is a "witness" in the above definitions. In our setting, where we consider algorithms induced by the outer approximation $\bar{\mathcal{Z}} \supseteq \mathcal{Z}$, algorithmic separability w.r.t. $\mathcal{A}$ is equivalent to separability w.r.t. $\bar{H} \triangleq \{\bar{h}_{\mathbf{w}} \mid \mathbf{w} \in \mathcal{W}\}$, where $\bar{h}_{\mathbf{w}}$ takes the form $\bar{h}_{\mathbf{w}}(x) = \arg\max_{\bar{\mathbf{z}}\in\bar{\mathcal{Z}}} \mathbf{w}^\top\mathbf{F}\bar{\mathbf{z}}$. Observe also that, in this setting, *algorithmic separability implies separability*; this was pointed out by Kulesza and Pereira (2007) and Finley and Joachims (2008). Essentially, their theoretical analyses for structured learning with outer approximate (or "overgenerating") inference algorithms are extensions of known results for the exact formulations, obtained by replacing $\mathcal{Z}$ with $\bar{\mathcal{Z}}$ in the formulas. While these theoretical guarantees are helpful (constrasting with undergenerating predictors, for which no guarantees exist), they do not provide sufficient conditions for algorithmic separability, neither do they bound the risk of an outer approximation compared with the exact formulation.

Lemma 1 *Assume that the feature function satisfies* $\|\mathbf{f}_r(x)\|_\infty \leq 1$,[2] *and let* $N_f \triangleq \max_{x\in\mathcal{X},r\in\mathcal{R}} \|\mathbf{f}_r(x)\|_1$. *Then, for any* $\mathbf{w} \in \mathcal{W}$, $\bar{\mathbf{z}} \in \bar{\mathcal{Z}}$ *and* $\mathbf{z} \in V(\mathcal{Z})$*:*

$$|\mathbf{w}^\top\mathbf{F}(\bar{\mathbf{z}} - \mathbf{z})| \quad\leq\quad \|\mathbf{w}\|_2\sqrt{N_f}\,\ell(\bar{\mathbf{z}};\mathbf{z}). \tag{7}$$

Proof: By using Hölder's inequality, $|\mathbf{w}^\top\mathbf{F}(\bar{\mathbf{z}} - \mathbf{z})| \leq \|\mathbf{F}^\top\mathbf{w}\|_\infty \cdot \|\bar{\mathbf{z}} - \mathbf{z}\|_1 = \max_{r\in\mathcal{R}} \mathbf{w}^\top\mathbf{f}_r(x) \cdot \ell(\bar{\mathbf{z}};\mathbf{z}) \leq \|\mathbf{w}\|_2\sqrt{N_f}\,\ell(\bar{\mathbf{z}};\mathbf{z})$. □

Note that Lemma 1 implies

$$\begin{aligned}\hat{\ell}(\bar{\mathbf{z}};\mathbf{z},\mathbf{w}) \quad&\triangleq\quad \mathbf{w}^\top\mathbf{F}(\bar{\mathbf{z}} - \mathbf{z}) + \ell(\bar{\mathbf{z}};\mathbf{z}) \\ &\leq\quad (1 + \|\mathbf{w}\|_2\sqrt{N_f})\,\ell(\bar{\mathbf{z}},\mathbf{z}).\end{aligned} \tag{8}$$

We can now provide a sufficient condition for algorithmic separability. Recall that $r_t(\mathbf{w}) \triangleq \max_{\mathbf{z}'_t\in\mathcal{Z}} \hat{\ell}(\mathbf{z}'_t;\mathbf{z}_t,\mathbf{w})$; defining analogously $\bar{r}_t(\mathbf{w}) \triangleq \max_{\bar{\mathbf{z}}'_t\in\bar{\mathcal{Z}}} \hat{\ell}(\bar{\mathbf{z}}'_t;\mathbf{z}_t,\mathbf{w})$, we have:

[2] This is guaranteed if the features are binary-valued, in which case N_f is the maximum possible number of active features at a single part.

Proposition 2 *Let L be such that, for any $\bar{\mathbf{z}} \in \bar{\mathcal{Z}} \setminus \mathcal{Z}$, there exists $\mathbf{z} \in V(\mathcal{Z})$ such that $\ell(\bar{\mathbf{z}}; \mathbf{z}) \leq L$. Then:*

$$r_t(\mathbf{w}) \;\leq\; \bar{r}_t(\mathbf{w}) \;\leq\; r_t(\mathbf{w}) + (1 + \|\mathbf{w}\|_2 \sqrt{N_f})L. \quad (9)$$

Furthermore, let $L' \geq L$ be such that, for any $\bar{\mathbf{z}} \in \bar{\mathcal{Z}} \setminus \mathcal{Z}$, there exist $\mathbf{z}, \mathbf{z}' \in V(\mathcal{Z})$ such that $\ell(\bar{\mathbf{z}}; \mathbf{z}) \leq \ell(\bar{\mathbf{z}}; \mathbf{z}') \leq L'$. If $\mathcal{L}$ is separated by $\mathbf{w}^*$ with a large margin (i.e. if $\sum_{t=1}^m r_t(\mathbf{w}^*) = 0$) and $\|\mathbf{w}^*\|_2 < 1/(L'\sqrt{N_f})$, then $\mathcal{L}$ is algorithmically separated by $\mathbf{w}^*$.

Proof: The fact that $r_t(\mathbf{w}) \leq \bar{r}_t(\mathbf{w})$ is trivial, since $\mathcal{Z} \subseteq \bar{\mathcal{Z}}$. As for the upper bound, we choose $\mathbf{z}'_t \in V(\mathcal{Z})$ such that $\ell(\bar{\mathbf{z}}'_t, \mathbf{z}'_t) \leq L$ and apply triangle inequality:

$$\begin{aligned}
\bar{r}_t(\mathbf{w}) \;&\leq\; \max_{\bar{\mathbf{z}}'_t \in \bar{\mathcal{Z}}} \mathbf{w}^\top \mathbf{F}_t(\bar{\mathbf{z}}'_t - \mathbf{z}'_t) + \mathbf{w}^\top \mathbf{F}_t(\mathbf{z}'_t - \mathbf{z}_t) \\
&\quad + \ell(\bar{\mathbf{z}}'_t; \mathbf{z}'_t) + \ell(\mathbf{z}'_t, \mathbf{z}_t) \\
&=\; \max_{\bar{\mathbf{z}}'_t \in \bar{\mathcal{Z}}} \hat{\ell}(\bar{\mathbf{z}}'_t; \mathbf{z}'_t, \mathbf{w}) + \hat{\ell}(\mathbf{z}'_t; \mathbf{z}_t, \mathbf{w}) \\
&\leq\; r_t(\mathbf{w}) + (1 + \|\mathbf{w}\|_2 \sqrt{N_f})L, \qquad (10)
\end{aligned}$$

As for the second part, we have for all t and $\bar{\mathbf{z}}'_t \in \bar{\mathcal{Z}}$:

$$\begin{aligned}
{\mathbf{w}^*}^\top \mathbf{F}_t(\mathbf{z}_t - \bar{\mathbf{z}}'_t) \;&=\; {\mathbf{w}^*}^\top \mathbf{F}_t(\mathbf{z}_t - \mathbf{z}'_t) + {\mathbf{w}^*}^\top \mathbf{F}_t(\mathbf{z}'_t - \bar{\mathbf{z}}'_t) \\
&\geq\; {\mathbf{w}^*}^\top \mathbf{F}_t(\mathbf{z}_t - \mathbf{z}'_t) - \|\mathbf{w}^*\|_2 \sqrt{N_f} \ell(\mathbf{z}'_t; \bar{\mathbf{z}}'_t) \\
&\geq\; \ell(\mathbf{z}_t; \mathbf{z}'_t) - \|\mathbf{w}^*\|_2 \sqrt{N_f} L' \\
&\geq\; 1 - 1 = 0, \qquad (11)
\end{aligned}$$

where we chose $\mathbf{z}'_t \neq \mathbf{z}_t$ such that $\ell(\bar{\mathbf{z}}'_t; \mathbf{z}'_t) \leq L'$, and used Lemma 1, together with the fact that any two distinct points in $V(\mathcal{Z})$ have at least unit loss (since they belong to $\mathbb{B}^n$). □

Corollary 3 *Under the conditions stated in the second part of Prop. 2, the perceptron algorithm running approximate inference will make a finite number of mistakes.*

Proof: (Sketch) Use Prop. 2 and the mistake bound for the structured perceptron (Collins, 2002). □

The bound (9) relies on a geometric characterization of the approximating polytope $\bar{\mathcal{Z}}$ (through the parameters L and L'). However, in some cases one has algorithmic approximation guarantees instead (see, e.g., Vazirani, 2001). The following establishes a bound similar to the one in Prop. 2 that relies on an algorithmic characterization (we define $\mathcal{A}(x, \mathbf{w}) \triangleq \bar{h}_{\mathbf{w}}(x)$):

Proposition 4 *If $\mathcal{A}$ is outer ϵ-approximate[3] for the class of problems $\min_{\mathbf{z} \in \mathcal{Z}} \mathbf{c}^\top \mathbf{z}$ with $\mathbf{c} \geq \mathbf{0}$, the vertices of $\mathcal{Z}$ have constant ℓ_1-norm (say K), and any $\bar{\mathbf{z}} \in \bar{\mathcal{Z}}$ satisfies $\mathbf{1}^\top \bar{\mathbf{z}} \leq K$, then:*

$$r_t(\mathbf{w}) \;\leq\; \bar{r}_t(\mathbf{w}) \;\leq\; r_t(\mathbf{w}) + 2\epsilon K(1 + \|\mathbf{w}\|_2 \sqrt{N_f}). \quad (12)$$

Proof: Since $\mathcal{A}$ is outer ϵ-approximate, it underestimates $\min_{\mathbf{z} \in \mathcal{Z}} \mathbf{c}^\top \mathbf{z}$ by at most $\epsilon \mathbf{c}^\top \mathbf{z}^*$, provided $\mathbf{c} \geq \mathbf{0}$. In the general case for LAI, however, $\mathbf{c} = -\mathbf{F}_t^\top \mathbf{w} - \mathbf{p}_t \not\geq \mathbf{0}$; but since the vertices of $\mathcal{Z}$ have constant norm, the problem of optimizing over $\mathcal{Z}$ is unchanged by adding any constant to the cost vector. Thus, defining $\mathbf{c}' = \mathbf{c} + \|\mathbf{c}\|_\infty \geq \mathbf{0}$:

$$\begin{aligned}
\bar{r}_t(\mathbf{w}) \;&=\; -\min_{\bar{\mathbf{z}}'_t \in \bar{\mathcal{Z}}} (\mathbf{c}' - \|\mathbf{c}\|_\infty \cdot \mathbf{1})^\top \bar{\mathbf{z}}'_t - \mathbf{w}^\top \mathbf{F}_t \mathbf{z}_t + q_t \\
&\leq\; -(1 - \epsilon) \min_{\mathbf{z}'_t \in \mathcal{Z}} {\mathbf{c}'}^\top \mathbf{z}'_t + K\|\mathbf{c}\|_\infty - \mathbf{w}^\top \mathbf{F}_t \mathbf{z}_t + q_t \\
&\leq\; r_t(\mathbf{w}) + \epsilon \min_{\mathbf{z}'_t \in \mathcal{Z}} {\mathbf{c}'}^\top \mathbf{z}'_t \\
&\leq\; r_t(\mathbf{w}) + 2\epsilon K(1 + \|\mathbf{w}\| \sqrt{N_f}),
\end{aligned}$$

again due to Hölder's inequality. □

Props. 2–4 will be used in Sec. 3.3 to establish empirical risk and generalization bounds.

3.2. Balancing Accuracy and Runtime

We now propose a new learning strategy that balances accuracy and algorithmic cost. We argue that, when the computational cost of performing inference *at test time* is something that we worry about, then this cost should be taken into account in the learning problem.

Let $\ell_c : H \times \mathcal{X} \to \mathbb{R}$ be a cost function that, given a hypothesis $h \in H$ and a data point $x \in \mathcal{X}$, expresses the cost of computing $h(x)$. Most existing learning algorithms concern minimizing the expected loss on unseen data, $\mathbb{E}\ell(h(X), Y)$; yet, it may happen that the model that minimizes this quantity (call it h^*) has an impractically high average computational cost. On the other hand, there may well exist another hypothesis $h' \in H$ performing similarly to h^* that yields much faster runtimes. Hence, our target should be to minimize $\mathbb{E}\left(\ell(h(X), Y) + \eta \ell_c(h, X)\right)$, where $\eta \geq 0$ is a trade-off parameter.

Traditional algorithmic complexity theory, which looks at the worst possible problem configurations, is not useful here, as we are interested in *average complexities* (Levin, 1986) under the unknown distribution $P(X, Y)$. As an example, consider the ILP formulation (6), where the cost vector $\mathbf{c} \triangleq \mathbf{F}^\top \mathbf{w}$ is affected both by the model parameters $\mathbf{w}$ and by the input data, represented in the matrix $\mathbf{F}$ (that we can see as a random variable). Although solving an ILP is an NP-complete problem, for some "nice" distributions $P(\mathbf{c})$ (peaked

[3] Given a class $\mathcal{F}$ of nonnegative functions and a minimization problem $\min_{x \in \mathcal{X}} f(x) \triangleq f^*$, an algorithm is said to be outer ϵ-approximate if it retrieves a lower bound of the optimum, $\underline{f}$, such that $(f^* - \underline{f})/f^* \leq \epsilon$.

over values of $\mathbf{c}$ that hit integer vertices of the constraint polyhedron) the average computational cost is low. Therefore, it is desirable to obtain model parameters $\mathbf{w}$ that, besides having small expected loss, yield nice cost vectors $\mathbf{c} \sim P(\mathbf{F}^\top \mathbf{w})$ with high probability.

When the constraint polytope does not contain integer points in its relative interior (which is our case, cf. (6)), the runtime of many off-the-shelf ILP solvers (like those based on branch-and-bounding or Gomory's cuts) decreases as the solution of the relaxed problem is closer to the exact solution; therefore, it may be reasonable to approximate the expected computational cost $\mathbb{E}\ell_c(h_{\mathbf{w}}, X)$ by the expected *relaxation gap* $\mathbb{E}\ell(h_{\mathbf{w}}(X), \bar{h}_{\mathbf{w}}(X))$. Led by this thought, we add a "empirical relaxation gap" term to our learning objective of the form $\frac{1}{m}\sum_{t=1}^m(\bar{r}_t(\mathbf{w}) - r_t(\mathbf{w})) \geq 0$. Rearranging terms, the overall learning problem becomes:

$$\min_{\mathbf{w}} \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1-\eta}{m}\sum_{t=1}^m r_t(\mathbf{w}) + \frac{\eta}{m}\sum_{t=1}^m \bar{r}_t(\mathbf{w}). \quad (13)$$

By defining $\tilde{\mathcal{Z}}_\eta \triangleq (1-\eta)\mathcal{Z} + \eta\bar{\mathcal{Z}}$ and due to linearity of the loss function, observe that each LAI problem in (13) can be written as $(1-\eta)r_t(\mathbf{w}) + \eta\bar{r}_t(\mathbf{w}) = \max_{\tilde{\mathbf{z}}'_t \in \tilde{\mathcal{Z}}_\eta} \mathbf{w}^\top \mathbf{F}_t(\tilde{\mathbf{z}}'_t - \mathbf{z}_t) + \ell(\tilde{\mathbf{z}}'_t, \mathbf{z}_t)$. As a consequence, the formulation (13) is isomorphic to the one in (5), with the sole difference that the polytope $\mathcal{Z}$ is replaced by $\tilde{\mathcal{Z}}_\eta$; unfortunately, $\tilde{\mathcal{Z}}_\eta$ is, in general, as hard to represent as $\mathcal{Z}$. Notice that $\mathcal{Z} \subseteq \tilde{\mathcal{Z}}_\eta \subseteq \bar{\mathcal{Z}}$ for any $\eta \in [0,1]$; geometrically, $\tilde{\mathcal{Z}}_\eta$ is obtained from $\bar{\mathcal{Z}}$ by intersecting the latter with cutting half-spaces that "carve out" the fractional vertices (see Fig. 1). Therefore, optimizing over $\tilde{\mathcal{Z}}_\eta$ has the effect of providing approximate solutions that lie near the integer vertices.

Rather than optimizing over the carved polytope, we propose a simple stochastic online strategy (see Algorithm 1) to tackle (13); we also allow the trade-off parameter to vary over time (i.e., we replace η by $\langle\eta_t\rangle_t$). This algorithm is similar to the subgradient algorithm of Ratliff et al. (2006), except that, at each step, it randomly decides whether it will perform exact or approximate LAI; we analyze this algorithm in Sec. 3.3.[4]

3.3. Generalization Bounds

Kulesza and Pereira (2007) observed that a PAC-Bayes generalization bound for empirical risk minimization can be adapted to the case of LP relaxed inference (by invoking the fact that relaxed inference essentially augments the output set). Here, we go farther and provide a generalization bound for approximate learning w.r.t. the empirical risk minimizer in the *exact* setting.

Algorithm 1 Modified Online Subgradient

Input: $\mathcal{L}$, $\langle\eta_t\rangle_t$, learning rate sequence $\langle\alpha_t\rangle_t$
Initialize $\mathbf{w}_1 \leftarrow \mathbf{0}$
for $t = 1$ **to** $m = |\mathcal{L}|$ **do**
  Pick $\sigma_t \sim$ Bernoulli(η_t)
  if $\sigma_t = 1$ **then**
    Solve relaxed LAI, $\hat{\mathbf{z}}_t \leftarrow \arg\max_{\bar{\mathbf{z}}'_t \in \bar{\mathcal{Z}}} \hat{\ell}(\bar{\mathbf{z}}'_t; \mathbf{z}_t, \mathbf{w}_t)$
  else
    Solve exact LAI, $\hat{\mathbf{z}}_t \leftarrow \arg\max_{\mathbf{z}'_t \in \mathcal{Z}} \hat{\ell}(\mathbf{z}'_t; \mathbf{z}_t, \mathbf{w}_t)$
  end if
  Compute the subgradient $\mathbf{g}_t \leftarrow \lambda\mathbf{w}_t + \mathbf{F}_t(\hat{\mathbf{z}}_t - \mathbf{z}_t)$
  Project and update $\mathbf{w}_{t+1} \leftarrow \mathrm{Proj}_{\mathcal{W}}(\mathbf{w}_t - \alpha_t\mathbf{g}_t)$
end for
Return the averaged model $\hat{\mathbf{w}} \leftarrow \frac{1}{m}\sum_{t=1}^m \mathbf{w}_t$.

Proposition 5 *Let $\hat{\mathbf{w}}$ be the solution returned by Algorithm 1 with learning rate chosen as $\alpha_t = 1/(\lambda t)$. Assume that $\hat{\ell}(.; \mathbf{z}_t, \mathbf{w}_t)$ is upper bounded by Λ and that the subgradient norm is bounded by G.[5] Then, the following bound holds with probability at least $1-\delta$:*

$$\mathbb{E}\ell(h_{\hat{\mathbf{w}}}(X); Y) \leq \min_{\mathbf{w}\in\mathcal{W}} \frac{1}{m}\sum_{t=1}^m r_t(\mathbf{w}) + M(\mathbf{w}, m) + \sqrt{8\Lambda^2/m \ln(2/\delta)}, \quad (14)$$

where $M(\mathbf{w}, m) \triangleq \lambda/2 \cdot \|\mathbf{w}\|^2 + G^2(1+\log m)/(2\lambda m) + (1 + \|\mathbf{w}\|\sqrt{N_f})L\sum_{t=1}^m \eta_t/m$.

Proof: (Sketch.) We adapt a result from Cesa-Bianchi et al. (2004) for convex and bounded loss functions to get (for any $0 < \delta' \leq 1$)

$$P\left(\mathbb{E}r(\hat{\mathbf{w}}) \geq \frac{1}{m}\sum_{t=1}^m r_t(\mathbf{w}_t) + \sqrt{\frac{2\Lambda^2}{m}\ln\frac{1}{\delta'}}\right) \leq \delta'. \quad (15)$$

Since $\ell(h_{\hat{\mathbf{w}}}(X); Y) \leq r(\hat{\mathbf{w}})$, it suffices to bound the RHS. Noting that $r_t(\mathbf{w}_t) \leq \bar{r}_t(\mathbf{w}_t)$, and adapting a regret bound from Ratliff et al. (2006) for the subgradient algorithm we get, for any $\mathbf{w} \in \mathcal{W}$:

[4] Note that Algorithm 1 with fixed η and fixed sample size indeed optimizes (13). The law of large numbers implies that after enough iterations, the fraction of time that each datum is used to solve exact LAI (resp. relaxed LAI) is arbitrarily close, in probability, to $1-\eta$ (resp. η); hence, one just needs to adapt the convergence proofs of the subgradient algorithm.

[5] The function $\hat{\ell}$ can be made bounded if we define $\mathcal{W}$ to be a convex body, e.g. by constraining $\|\mathbf{w}\| \leq \rho$, which ensures, through Lemma 1, that $\hat{\ell}(.; \mathbf{z}_t, \mathbf{w}_t) \leq (1+\rho N_f^{1/2})K \triangleq \Lambda$, where $K \geq \|\mathbf{z}\|_1$. As for G, assuming feature vectors whose norm is bounded by $R/2$, we may take $G = R + \lambda\rho^2$.

$$\frac{1}{m}\sum_{t=1}^{m} r_t(\mathbf{w}_t) \leq \frac{1}{m}\sum_{t=1}^{m} r_t(\mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{G^2(1+\log m)}{2\lambda m} + \frac{1}{m}\sum_{t=1}^{m} \sigma_t(\bar{r}_t(\mathbf{w}) - r_t(\mathbf{w})); \quad (16)$$

Now, applying Hoeffding's inequality to the random sequence $\langle \sigma_t \rangle_{t=1,\ldots,m}$, we get (for any $0 < \delta'' \leq 1$) $P(\sum_{t=1}^{m} \sigma_t \geq \sum_{t=1}^{m} \eta_t + \sqrt{2/m \cdot \log(1/\delta'')}) \leq \delta''$. Substituting in (16), invoking Prop. 2, and plugging the result in (15) (noting that Λ upper bounds the approximation gap), we get the desired result (by taking $\delta \triangleq 2\delta'' = 2\delta'$). □

Corollary 6 *If we set* $\lambda = \Theta(\sqrt{(1+\log m)/m})$ *and* $\eta_t = \Theta(1/\sqrt{t})$*, then* $\mathbb{E}r(\hat{\mathbf{w}}) \xrightarrow{a.s.} \frac{1}{m}\sum_{t=1}^{m} r_t(\mathbf{w}^*)$.
Proof: The choice of λ was proposed by Ratliff et al., 2006. As for $\langle \eta_t \rangle$, we have $\sum_{t=1}^{m} 1/\sqrt{t} = O(\sqrt{m}) = o(m)$; therefore $\lim_{m\to\infty} M(\mathbf{w}, m) = 0$. □

4. Dependency Parsing

We next show how approximate learning using LP relaxed inference can be used in an important NLP task involving non-local interactions among output variables: dependency parsing. We merely sketch the problem; see Martins et al. (2009) for a full discussion of dependency parsing and its ILP representations.

Dependency trees are a lightweight syntactic representation that attempts to capture functional relationships between words. Given a sentence $x = \langle w_0, \ldots, w_n \rangle$ (where w_i denotes the word at the i-th position, and $w_0 = \$$ is a wall symbol), consider the digraph $D = (V, A)$, with vertices in $V = \{0, \ldots, n\}$ (the i-th vertex corresponding to the i-th word) and arcs in $A = V^2$. A (legal) dependency parse tree of x is any 0-arborescence[6] of D; we denote the set of legal dependency parse trees of x by $\mathcal{Y}(x)$.[7] If $a = (i, j) \in y$ we refer to i as the parent of j and j as the child of i.

A **parser** is a function $h : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{Y} = \bigcup_{x\in\mathcal{X}} \mathcal{Y}(x)$. The fact that $\mathcal{Y}(x)$ is exponentially large makes this a structured classification problem. We want to learn an h with small expected loss—here, the Hamming loss function $\ell(y'; y) \triangleq |\{(i, j) \in y' : (i, j) \notin y\}|$, which, if $\mathcal{R} = A$, is proportional to (3).

[6] An r-arborescence of D is a subset B of A such that (V, B) is a (directed) tree rooted at r.

[7] In this paper, we consider *unlabeled* dependency parsing, where only the backbone structure is predicted.

An **arc-factored** model for dependency parsing is one in which the feature vector decomposes as a sum over arcs (i.e., $\mathcal{R} = A$ in (1)); such models can capture each word's preferences for particular properties of its children or parents. McDonald et al. (2005) showed that arc-factored inference among trees is an instance of the *minimum arborescence problem*, which enables efficient algorithms for exact inference (Chu & Liu, 1965; Edmonds, 1967).

Riedel and Clarke (2006) added hard linguistic constraints to an arc-factored model, representing the inference problem as an ILP with exponentially many constraints. They used a cutting plane algorithm for inference, in which constraints are only invoked when violated; further, they trained with an arc-factored model since the cutting plane algorithm was slow. In Martins et al. (2009), we formulate dependency parsing as an ILP with a *polynomial* number of constraints, by adapting a single-commodity directed flow model due to Magnanti and Wolsey (1994). Our representation allows constraints to be made *soft*, so that their strengths are *learned* as features of the model. This permits us to include **non-arc-factored** features, described next.

Siblings and grandparents It was shown by McDonald et al. (2006) and Smith and Eisner (2008) that modeling interactions among words who share a parent or among a word's children and its parent can be beneficial. To incorporate these features, we employ a linearization trick (Boros & Hammer, 2002). This can be done with $O(n^3)$ variables and constraints.

Valency Words in a language have preferences not only for which words will be their children, but also *how many* children they will have (valency or arity). Our model includes valency indicator features. An extra $O(n^2)$ variables and constraints are necessary.

Projectivity If $y \in \mathcal{Y}(x)$, we say that an arc $a = (i, j) \in y$ is **projective** if for any vertex k in the span of a (i.e. satisfying $\min(i, j) < k < \max(i, j)$), there exists a path in y from i to k. A dependency tree is called projective if it only contains projective arcs.[8] Although non-projectivity is arguably necessary for correctly capturing dependency structure in some languages, parse trees tend to be nearly projective. We encode this preference in a learned, language-specific way, as a feature. Indicator variables for projective arcs can be added with an extra $O(n^3)$ variables and constraints.

[8] When the arc-factored assumption is weakened and non-projectivity is permitted, exact inference becomes NP-hard (McDonald & Satta, 2007), cf. parsing with non-projectivity disallowed (Eisner, 1996).

Table 1. Results for dependency parsing. We have reproduced the system of McDonald et al. (2006) for the sake of comparison (MLP'06). For each language and model setting, we report the unlabeled attachment scores (UAS, %) using exact and approximate inference at test time. For the arc-factored model, we report the results obtained by learning with exact LAI. Bold indicates significantly best results (statistical significance is measured by Dan Bikel's randomized parsing evaluation comparator, `http://www.cis.upenn.edu/~dbikel/software.html`).

	MLP'06	Arc-Factored Model				Full Model	
Learning →		Exact ($\eta = 0$)		Approx. ($\eta = 1$)		Approx. ($\eta = 1$)	
Inference →		Exact	Approx.	Exact	Approx.	Exact	Approx.
Danish	90.60	89.86	89.68	89.80	89.78	**91.18**	**91.04**
Dutch	84.11	83.15	83.17	83.55	83.61	**85.57**	**85.41**
Portuguese	**91.40**	90.66	90.66	90.66	90.70	**91.42**	**91.44**
Slovene	83.67	84.05	83.87	83.93	83.95	**85.61**	**85.41**

5. Experiments

We demonstrate the effectiveness of our polyhedral approximations for dependency parsing, with experiments on four languages from the CoNLL-X shared task (Buchholz & Marsi, 2006): Danish, Dutch, Portuguese and Slovene. We used the same arc-factored features as McDonald et al. (2005) and optional non-arc-factored features as described in Sec. 4. All our experiments were conducted on a PC with a Intel dual-core processor with 2.66 GHz and 2 Gb RAM memory. We used CPLEX to solve the ILPs.

For scalability, we first prune the base graph by running a simple algorithm that ranks the k-best candidate parents for each word in the sentence, setting $k = 10$; this reduces the number of variables and constraints in the arc-factored model to $O(nk)$, and in the full model to $O(n^2k)$.[9] The ranker is a local model trained using a max-margin criterion; it is arc-factored and not subject to *any* structural constraints, so it is fast. Pruning was employed in both training and testing. To learn the actual parser, we implemented Alg. 1 with passive-aggressive updates (Crammer et al., 2006).[10] At test time, we experimented with exact and approximate inference. The approximate decoder was implemented as follows to obtain a true parse: first, solve the LP relaxation; then, if the solution $\mathbf{z}^*$ is fractional, project its arc components $\mathbf{z}^*_A \triangleq (\mathbf{z}^*_a)_{a \in A}$ onto the feasible set $\mathcal{Y}(x)$. The Euclidean projection can be computed by finding a maximal arborescence in the digraph whose weights are defined by $\mathbf{z}^*_A$ (proof omitted); as seen in Sec. 4, the Chu-Liu-Edmonds algorithm can do this in polynomial time.

Tab. 1 summarizes the results. As expected, adding non-arc-factored features makes the models more accurate. We also observe that approximate training did not hurt the arc-factored model, compared with exact training. Moreover, approximate decoding at test time did not considerably affect accuracy for any of the models. For 3 out of 4 languages, the full model yields substantially better results than the approximate non-arc-factored parser of McDonald et al. (2006).

[9] The oracle constrained to pick parents from these lists achieves $> 98\%$ in every case.

[10] Without regularization, this can be seen as a variant of the online subgradient algorithm of Ratliff et al. (2006).

Table 2. Runtimes for Slovene, as a function of η. Learning the full model was intractable as $\eta \to 0$; the valency and non-projectivity features were excluded in this analysis. The reported values (collected at test time) are: the percentage of *words* for which a fractional parent (FP) was assigned in the LP-relaxed problem, and average runtimes per sentence, using exact and approximate inference.

η	FP (%)	Exact (sec.)	Approx. (sec.)
0.00	16.53	19.33	0.17
0.25	10.42	4.16	0.13
0.50	8.29	2.20	0.13
0.75	7.76	1.62	0.13
1.00	7.12	1.11	0.12

To see whether the parameter η is effectively penalizing computational cost, according to the paradigm sketched in Sec. 3.2, we did an additional experiment for the Slovene dataset (Tab. 2). We observe that, as η approaches 0 (i.e., as training becomes close to exact), the learned model tends to assign more fractional solutions in the LP relaxed problem (a subroutine for both the approximate and exact decoders), which results in a dramatic increase in runtime for the exact decoder. In contrast, when trained with $\eta = 1$, the model learns to avoid fractional solutions, and ILP solvers will converge faster to the optimum (on average). Yet the approximate decoder is still significantly faster.

6. Conclusions

We studied the impact of LP relaxed inference in max-margin learning. Based on a geometric characterization, we established conditions that guarantee algorithmic separability and derived risk bounds w.r.t. the exact formulations. As a by-product, we put forth a new learning paradigm that takes computational cost into consideration. We demonstrated the effectiveness

of these techniques on a structured prediction problem. As future work, we will look for polyhedral characterizations that guarantee tighter risk bounds; in particular, we aim to obtain conditions under which the approximation gap decreases with the sample size.

Acknowledgments

The authors thank the reviewers for their comments and Nathan Ratliff for discussions. A.M. was supported by a grant from FCT/ICTI through the CMU-Portugal Program, and also by Priberam Informática. N.S. was supported by NSF IIS-0836431 and an IBM Faculty Award. E.X. was supported by NSF DBI-0546594, DBI-0640543, IIS-0713379, and an Alfred Sloan Foundation Fellowship in Computer Science.

References

Boros, E., & Hammer, P. (2002). Pseudo-Boolean optimization. *Discrete Applied Math.*, *123*, 155–225.

Buchholz, S., & Marsi, E. (2006). CoNLL-X shared task on multilingual dependency parsing. *Conf. Comp. Nat. Lang. Learn.*, 189–210.

Cesa-Bianchi, N., Conconi, A., & Gentile, C. (2004). On the generalization ability of on-line learning algorithms. *IEEE Trans. on Inf. Theory*, *50*, 2050–2057.

Chang, M., Ratinov, L., & Roth, D. (2008). Constraints as prior knowledge. *Int. Conf. Mach. Learn. Workshop on Prior Knowledge for Text and Language Processing*.

Chu, Y. J., & Liu, T. H. (1965). On the shortest arborescence of a directed graph. *Science Sinica*, *14*, 1396–1400.

Clarke, J., & Lapata, M. (2008). Global inference for sentence compression: An integer linear programming approach. *J. Art. Intel. Res.*, *31*, 399–429.

Collins, M. (2002). Discriminative training methods for HMMs: Theory and experiments with perceptron algorithms. *Emp. Meth. Nat. Lang. Proc.*

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., & Singer, Y. (2006). Online passive-aggressive algorithms. *J. Mach. Learn. Res.*, *7*, 551–585.

Daumé, H., & Marcu, D. (2005). Learning as search optimization: Approximate large margin methods for structured prediction. *Int. Conf. Mach. Learn.*, 169–176.

Denis, P., & Baldridge, J. (2007). Joint determination of anaphoricity and coreference resolution using integer programming. *North. Am. Assoc. Comp. Ling.*

Edmonds, J. (1967). Optimum branchings. *J. Res. National Bureau of Standards*, *71B*, 233–240.

Eisner, J. (1996). Three new probabilistic models for dependency parsing: An exploration. *Int. Conf. Comp. Ling.*, 340–345.

Finley, T., & Joachims, T. (2008). Training structural SVMs when exact inference is intractable. *Int. Conf. Mach. Learn.*, 304–311.

Kulesza, A., & Pereira, F. (2007). Structured learning with approximate inference. *Neur. Inf. Proc. Sys. 20*, 785–792.

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Int. Conf. Mach. Learn.*, 282–289.

Levin, L. (1986). Average case complete problems. *SIAM Journal on Computing*, *15*, 285.

Magnanti, T., & Wolsey, L. (1994). *Optimal Trees* (Tech. Rep. 290-94). MIT, Op. Res. Center.

Martins, A. F. T., Smith, N. A., & Xing, E. P. (2009). Concise integer linear programming formulations for dependency parsing. *Assoc. Comp. Ling.* To appear.

McDonald, R., Lerman, K., & Pereira, F. (2006). Multilingual dependency analysis with a two-stage discriminative parser. *Conf. Comp. Nat. Lang. Learn.*, 216–220.

McDonald, R., & Satta, G. (2007). On the complexity of non-projective data-driven dependency parsing. *Int. Conf. Pars. Tech.*, 121–132.

McDonald, R. T., Pereira, F., Ribarov, K., & Hajic, J. (2005). Non-projective dependency parsing using spanning tree algorithms. *Emp. Meth. Nat. Lang. Proc.*, 523–530.

Ratliff, N., Bagnell, J., & Zinkevich, M. (2006). Subgradient methods for maximum margin structured learning. *Int. Conf. Mach. Learn. Workshop on Learning in Structured Outputs Spaces*.

Richardson, M., & Domingos, P. (2006). Markov logic networks. *Mach. Learn.*, *62*, 107–136.

Riedel, S., & Clarke, J. (2006). Incremental integer linear programming for non-projective dependency parsing. *Emp. Meth. Nat. Lang. Proc.*, 129–137.

Roth, D., & Yih, W. (2005). Integer linear programming inference for conditional random fields. *Int. Conf. Mach. Learn.*, 736–743.

Smith, D. A., & Eisner, J. (2008). Dependency parsing by belief propagation. *Emp. Meth. Nat. Lang. Proc.*, 145–156.

Taskar, B., Guestrin, C., & Koller, D. (2004a). Max-margin markov networks. *Neur. Inf. Proc. Sys. 16*.

Taskar, B., Chatalbashev, V., & Koller, D. (2004b). Learning associative Markov networks. *Int. Conf. Mach. Learn.*

Taskar, B., Lacoste-Julien, S., & Jordan, M. (2006). Structured prediction via the extragradient method. *Neur. Inf. Proc. Sys. 18*, 1345–1352.

Tsochantaridis, I., Hofmann, T., Joachims, T., & Altun, Y. (2004). Support vector machine learning for interdependent and structured output spaces. *Int. Conf. Mach. Learn.*

Vazirani, V. (2001). *Approximation Algorithms*. Springer.