On Robustness and Regularization of Structural Support Vector Machines

MohamadAli Torkamani ALI@CS.UOREGON.EDU
Daniel Lowd LOWD@CS.UOREGON.EDU
Computer and Information Science Department, University of Oregon

Abstract

Previous analysis of binary support vector machines (SVMs) has demonstrated a deep connection between robustness to perturbations over uncertainty sets and regularization of the weights. In this paper, we explore the problem of learning robust models for structured prediction problems. We first formulate the problem of learning robust structural SVMs when there are perturbations in the sample space, and show how we can construct corresponding bounds on the perturbations in the feature space. We then show that robustness to perturbations in the feature space is equivalent to additional regularization. For an ellipsoidal uncertainty set, the additional regularizer is based on the dual norm of the norm that constrains the ellipsoidal uncertainty. For a polyhedral uncertainty set, the robust optimization problem is equivalent to adding a linear regularizer in a transformed weight space related to the linear constraints of the polyhedron. We also show that these constraint sets can be combined and demonstrate a number of interesting special cases. This represents the first theoretical analysis of robust optimization of structural support vector machines. Our experimental results show that our method outperforms the nonrobust structural SVMs on real world data when the test data distribution has drifted from the training data distribution.

1. Introduction

Traditional machine learning methods assume that training and test data are drawn from the same distribution. However, in many real-world applications, the distribution is constantly changing. In some cases, such as spam filtering and fraud detection, an adversary may be actively manipulating it to defeat the learned model. In others, such as news and political discussions, the concept changes quickly over time and we want to be robust to these unpredictable changes. In both scenarios, it is beneficial to optimize the model's performance on not just the training data but on the worst-case manipulation of the training data, where the manipulations are constrained to some domain-specific uncertainty set. For example, in an image classification problem, the uncertainty set could include minor translations, rotations, noise, or color shifts of the training data. This type of robust optimization leads to models that perform well on points that are "close" to those in the training data.

In general, robust optimization addresses optimization problems in which some degree of uncertainty governs the known parameters of the model (Ben-Tal & Nemirovski, 1998; 1999; 2000; 2001; Bertsimas & Sim, 2004). Robust linear programming is central to many of the existing formulations. For example, Bertsimas et al. (2004) show that when the disturbance of the inputs is restricted to an ellipsoid around the true values defined by some norm, then the robust linear programming problem can be reduced to a convex cone program. A number of other authors have explored the application of robust optimization to classification problems (e.g., (Lanckriet et al., 2003; El Ghaoui et al., 2003; Bhattacharyya et al., 2004; Shivaswamy et al., 2006)). Recently, Xu et al. (2009) showed that regularization of support vector machines (SVMs) can be derived from a robust formulation. However, robustness for structured prediction models has remained largely unexplored. Structured prediction problems are characterized by an exponentially large space of possible outputs, such as parse trees or graph labelings, making this a much more challenging problem.

In this paper, we develop a general-purpose technique for learning robust structural SVMs (Tsochantaridis et al., 2004). Our basic approach is to consider the worst-case corruption of the input data within some uncertainty set and use this to define a robust formulation. This optimization problem is often much harder than standard training of structural SVMs when written directly; we overcome this obstacle by transforming the robust optimization problem into a standard structural SVM learning problem with an additional regularizer. This gives us both robustness and computational efficiency in the structured prediction setting, as well as establishing an elegant relationship between robustness and regularization for structural SVMs.

We demonstrate our approach on a new dataset consisting of snapshots of political blogs from 2003 through

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32. Copyright 2014 by the author(s).

2013, based on the political blogs dataset from Adamic and Glance (2005). Blogs are classified as liberal or conservative using both their words and link structure. To make this more challenging, we train on blogs from 2004 but evaluate on every year, from 2003 to 2013. In this domain, we define an uncertainty set, show how to construct an appropriate regularizer, and show that this regularization can lead to substantially lower test error than a non-robust model.

2. Notation and Background

$\boldsymbol{x}$ and $\boldsymbol{y}$ denote the vectorized input and the representation of the structured output in the training data, respectively. For simplicity of notation, we assume a single training example, such as a single social network graph, but our results easily extend to a set of training examples.

The feature vector $\boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{y})$ is a function of both inputs and outputs (and also manipulated input or alternate outputs, when used as the input argument). We use $\boldsymbol{\Delta\phi}(\boldsymbol{x}, \boldsymbol{y}, \tilde{\boldsymbol{y}})$ to refer to the difference between two feature vectors with different outputs $\boldsymbol{y}$ and $\tilde{\boldsymbol{y}}$:

$$\boldsymbol{\Delta\phi}(\boldsymbol{x}, \boldsymbol{y}, \tilde{\boldsymbol{y}}) \equiv \boldsymbol{\phi}(\boldsymbol{x}, \tilde{\boldsymbol{y}}) - \boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{y}).$$

The value of $\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}, \tilde{\boldsymbol{y}})$ is called the *score* of labeling $\boldsymbol{x}$ as $\tilde{\boldsymbol{y}}$, for the given model weights $\boldsymbol{w}$.

$\Delta(\boldsymbol{y}, \tilde{\boldsymbol{y}})$ is a scalar distance function, such as Hamming distance, which is a measure of dissimilarity between the true and alternate outputs.

We use $\|.\|$ to refer to a general norm function and $\|.\|^*$ for the dual norm of $\|.\|$, where $\|\boldsymbol{y}\|^* = \sup\{\boldsymbol{y}^T\boldsymbol{x} | \|\boldsymbol{x}\| \leq 1\}$.

In this paper, we focus on the derivation of robust formulations for 1-slack structural SVM (Joachims et al., 2009). (With minor changes, the results of this paper can be applied to n-slack structural SVMs as well, but we skip them here.) The optimization program of a 1-slack structural SVM is:

$$\begin{aligned} &\underset{\boldsymbol{w},\zeta}{\text{minimize}}\ f(\boldsymbol{w}) + C\zeta \quad \text{subject to} \\ &\zeta \geq \max_{\tilde{\boldsymbol{y}}}\ \boldsymbol{w}^T\boldsymbol{\Delta\phi}(\boldsymbol{x}, \boldsymbol{y}, \tilde{\boldsymbol{y}}) + \Delta(\boldsymbol{y}, \tilde{\boldsymbol{y}}) \end{aligned} \tag{1}$$

where $\boldsymbol{x}$ is the vector of all input variables, $\boldsymbol{y}$ is the desired structured query variables, and $\boldsymbol{w}$ is the vector of the model parameters. The goal is to learn $\boldsymbol{w}$.

$f(\boldsymbol{w})$ is a regularization function that penalizes "large" weights. Depending on the application, $f(\boldsymbol{w})$ can be any convex function in general. Semi-homogeneous functions, such as norms or powers of norms with power value equal to or greater than 1, are a common choice. (A function $f(z)$ is semi-homogeneous if and only if $f(az) = a^\alpha f(z)$ for some positive α.) $f(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$ is the most commonly used regularization function.

3. Robust Structural SVMs

In this section, we motivate and define a robust formulation of structural SVMs. We begin by considering how an adversary might modify an input in order to maximize the prediction error, and use this to derive a definition of a robust structural SVM in sample space and feature space.

3.1. Worst-Case/Adversarial Data Manipulation

Adversaries might have a wide range of goals, but in the worst case they will antagonistically try to reduce the accuracy of the predictive model. For structural SVMs, the predicted output is chosen by solving $\tilde{\boldsymbol{y}} = \arg\max_{\tilde{\boldsymbol{y}}} \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}, \tilde{\boldsymbol{y}})$, where $\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x}, \tilde{\boldsymbol{y}})$ is the classification score. Thus, an adversary's antagonistic goal would be to replace the true input $\boldsymbol{x}$ with a manipulated version $\tilde{\boldsymbol{x}}$ that maximizes the classification loss $\Delta(\boldsymbol{y}, \tilde{\boldsymbol{y}})$. If the highest scoring label is not unique, we assume the adversary tries to maximize the minimum loss in the set:

$$\begin{aligned} &\underset{\tilde{\boldsymbol{x}}}{\text{maximize}}\ \min_{\tilde{\boldsymbol{y}}}\ \Delta(\boldsymbol{y}, \tilde{\boldsymbol{y}}), \quad \text{subject to} \\ &\tilde{\boldsymbol{y}} \in \arg\max_{\tilde{\boldsymbol{y}} \neq \boldsymbol{y}} \boldsymbol{w}^T\boldsymbol{\phi}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) \\ &\tilde{\boldsymbol{x}} \in \mathcal{S}(\boldsymbol{x}, \boldsymbol{y}) \end{aligned} \tag{2}$$

$\mathcal{S}(\boldsymbol{x}, \boldsymbol{y})$ is a domain-specific *uncertainty set*, which constrains the set of possible corrupt inputs $\tilde{\boldsymbol{x}}$. We always assume that $\boldsymbol{x} \in \mathcal{S}(\boldsymbol{x}, \boldsymbol{y})$, which means $\boldsymbol{x}$ can remain unchanged. The set $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y})$ can contain a wide range of possible variations, such as the amount of affordable/possible change in an attribute, or the restrictions that are enforced on combinations of changes among several attributes.

The bi-level optimization program in (2) is not tractable in general, especially when $\boldsymbol{x}$ and $\boldsymbol{y}$ have integer components. A slightly more tractable solution is to relax the program and only require that $\tilde{\boldsymbol{y}}$ be scored higher than the true output $\boldsymbol{y}$:

$$\begin{aligned} &\underset{\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}}{\text{maximize}}\ \Delta(\boldsymbol{y}, \tilde{\boldsymbol{y}}), \quad \text{subject to} \\ &\boldsymbol{w}^T\boldsymbol{\phi}(\tilde{\boldsymbol{x}}, \boldsymbol{y}) \leq \boldsymbol{w}^T\boldsymbol{\phi}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) \\ &\tilde{\boldsymbol{x}} \in \mathcal{S}(\boldsymbol{x}, \boldsymbol{y}) \end{aligned} \tag{3}$$

The maximization in (3) might be infeasible, but its Lagrangian relaxation is always feasible:

$$\begin{aligned} &\underset{\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}}{\text{maximize}}\ \lambda\boldsymbol{w}^T\boldsymbol{\Delta\phi}(\tilde{\boldsymbol{x}}, \boldsymbol{y}, \tilde{\boldsymbol{y}}) + \Delta(\boldsymbol{y}, \tilde{\boldsymbol{y}}) \\ &\text{subject to} \quad \tilde{\boldsymbol{x}} \in \mathcal{S}(\boldsymbol{x}, \boldsymbol{y}) \end{aligned} \tag{4}$$

We want to attract the reader's attention to the similarity of (4), and the nested max operation in the constraint of (1). In fact, $\lambda\boldsymbol{w}^T\boldsymbol{\Delta\phi}(\tilde{\boldsymbol{x}}, \boldsymbol{y}, \tilde{\boldsymbol{y}}) + \Delta(\boldsymbol{y}, \tilde{\boldsymbol{y}})$ is a component of the loss function that the learner wants to minimize. In the next subsection, we reformulate the standard 1-slack structural SVM so that the effect of adversarial manipulation of input data will be minimized.

3.2. Robust Formulation in Sample Space

Our goal is to find a set of model parameters that perform well against the worst-case manipulated input $\tilde{\boldsymbol{x}}$ in the uncertainty set. We formulate this by replacing the loss-augmented margin in (1) with the worst-case adversarial loss obtained by (4):

$$\underset{\boldsymbol{w}}{\text{minimize}}\, Cf(\boldsymbol{w}) + \sup_{\tilde{\boldsymbol{x}}\in\mathcal{S}(\boldsymbol{x},\boldsymbol{y}),\tilde{\boldsymbol{y}}} \mathcal{L}_\lambda(\boldsymbol{w},\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}},\boldsymbol{y}) \quad (5)$$

where $\mathcal{L}_\lambda(\boldsymbol{w},\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}},\boldsymbol{y}) = \lambda\boldsymbol{w}^T\boldsymbol{\Delta\phi}(\tilde{\boldsymbol{x}},\boldsymbol{y},\tilde{\boldsymbol{y}})+\Delta(\boldsymbol{y},\tilde{\boldsymbol{y}})$. We replace the maximization with a sup operator to indicate that the maximum value might not be achieved. Both λ and C are tunable parameters that can be determined by cross-validation. In the following lemma we show that it is possible to tune only one of them by performing a re-parameterization.

Lemma 3.1. *For semi-homogeneous $f(.)$, the problem (5) can be equivalently re-written in the following form:*

$$\underset{\boldsymbol{w}}{\text{minimize}}\, Cf(\boldsymbol{w}) + \sup_{\tilde{\boldsymbol{x}}\in\mathcal{S}(\boldsymbol{x},\boldsymbol{y}),\tilde{\boldsymbol{y}}} \mathcal{L}(\boldsymbol{w},\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}},\boldsymbol{y}) \quad (6)$$

where $\mathcal{L}(\boldsymbol{w},\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}},\boldsymbol{y}) = \boldsymbol{w}^T\boldsymbol{\Delta\phi}(\tilde{\boldsymbol{x}},\boldsymbol{y},\tilde{\boldsymbol{y}})+\Delta(\boldsymbol{y},\tilde{\boldsymbol{y}})$

Proof. Let $\boldsymbol{w}' = \lambda\boldsymbol{w}$, and $C' = \frac{C}{\lambda^\alpha}$. Then, for a semi-homogeneous $f(.)$, where $f(a\boldsymbol{w}) = a^\alpha f(\boldsymbol{w})$, we have $Cf(\boldsymbol{w}) = \frac{C}{\lambda^\alpha}f(\lambda\boldsymbol{w})$. Therefore, by re-parameterization of $\boldsymbol{w}$ as $\boldsymbol{w}'$ and C as C', (5) can be rewritten as (6). □

Problem (6) is similar in form to a standard structural SVM, except that the inner maximization is done over both $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{y}}$. This is potentially much harder than simply maximizing over $\tilde{\boldsymbol{y}}$, since the input often has a much higher dimension than the output. For example, when labeling a set of 1000 web pages, there are only 1000 labels to predict but 1,000,000 possible hyperlinks that the adversary could add or remove. In the next subsection, we show that we can avoid the above-mentioned computational complexity by instead restricting the variations in the feature space.

3.3. Robustness in Feature Space

Let $\boldsymbol{\Delta x}$ be the disturbance in the sample space such that: $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\Delta x}$. Then, by finite difference approximation[1]:

$$\begin{aligned}\boldsymbol{\phi}(\tilde{\boldsymbol{x}},\boldsymbol{y}) &= \boldsymbol{\phi}(\boldsymbol{x}+\boldsymbol{\Delta x},\boldsymbol{y}) = \boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{y}) + \boldsymbol{\delta}(\tilde{\boldsymbol{x}},\boldsymbol{y})\\ \boldsymbol{\phi}(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}) &= \boldsymbol{\phi}(\boldsymbol{x}+\boldsymbol{\Delta x},\tilde{\boldsymbol{y}}) = \boldsymbol{\phi}(\boldsymbol{x},\tilde{\boldsymbol{y}}) + \boldsymbol{\delta}(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})\end{aligned}$$

Note that we are not introducing any error; both functions $\boldsymbol{\delta}(\tilde{\boldsymbol{x}},\boldsymbol{y})$ and $\boldsymbol{\delta}(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})$ contain as many high-order approximation terms as needed for achieving infinitesimal error introduction, although we never unpack these functions. In fact, the difference between $\boldsymbol{\delta}(\tilde{\boldsymbol{x}},\boldsymbol{y})$ and $\boldsymbol{\delta}(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}})$ is particularly important; let $\boldsymbol{\delta}_{\tilde{\boldsymbol{y}}}(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{x}}) = \boldsymbol{\delta}(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}) - \boldsymbol{\delta}(\tilde{\boldsymbol{x}},\boldsymbol{y})$, then:

$$\begin{aligned}&\boldsymbol{\phi}(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}) - \boldsymbol{\phi}(\tilde{\boldsymbol{x}},\boldsymbol{y})\\ =\ &\boldsymbol{\phi}(\boldsymbol{x}+\boldsymbol{\Delta x},\tilde{\boldsymbol{y}}) - \boldsymbol{\phi}(\boldsymbol{x}+\boldsymbol{\Delta x},\boldsymbol{y})\\ =\ &\boldsymbol{\phi}(\boldsymbol{x},\tilde{\boldsymbol{y}}) - \boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{y}) + \boldsymbol{\delta}(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}) - \boldsymbol{\delta}(\tilde{\boldsymbol{x}},\boldsymbol{y})\\ =\ &\boldsymbol{\Delta\phi}(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{y}}) + \boldsymbol{\delta}_{\tilde{\boldsymbol{y}}}(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{x}}) \qquad (7)\end{aligned}$$

Therefore, the manipulation of the input data affects the margin $\mathcal{L}(.)$ in (6) through $\boldsymbol{\delta}_{\tilde{\boldsymbol{y}}}(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{x}})$. In the rest of the paper, we will use $\boldsymbol{\delta}^i$ to refer to the ith element of the vector $\boldsymbol{\delta}_{\tilde{\boldsymbol{y}}}(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{x}})$.

Clearly, $\boldsymbol{\delta}_{\tilde{\boldsymbol{y}}}$ depends on the specific choice of the alternate labeling $\tilde{\boldsymbol{y}}$, as well as $\tilde{\boldsymbol{x}}$, $\boldsymbol{x}$, and $\boldsymbol{y}$. Let $\Delta^2\Phi(\boldsymbol{x},\boldsymbol{y})$ be the set of all variations, over all $\tilde{\boldsymbol{y}}$ and $\tilde{\boldsymbol{x}}$:

$$\Delta^2\Phi(\boldsymbol{x},\boldsymbol{y}) \equiv \{\boldsymbol{\delta} = \boldsymbol{\delta}_{\tilde{\boldsymbol{y}}}(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{x}}) |\ \forall\tilde{\boldsymbol{x}}\in\mathcal{S}(\boldsymbol{x},\boldsymbol{y}),\tilde{\boldsymbol{y}}\}$$

Note that $\Delta^2\Phi(\boldsymbol{x},\boldsymbol{y})$ is independent of $\tilde{\boldsymbol{y}}$. In the next section, we introduce some mechanical procedures for calculating $\Delta^2\Phi(\boldsymbol{x},\boldsymbol{y})$ from $\mathcal{S}(\boldsymbol{x},\boldsymbol{y})$, for certain choices of $\mathcal{S}(\boldsymbol{x},\boldsymbol{y})$ and $\boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{y})$.

Lemma 3.2. *Let $\mathcal{L}_1(\boldsymbol{w},\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}},\boldsymbol{y}) = \boldsymbol{w}^T\boldsymbol{\Delta\phi}(\tilde{\boldsymbol{x}},\boldsymbol{y},\tilde{\boldsymbol{y}}) + \Delta(\boldsymbol{y},\tilde{\boldsymbol{y}})$, and $\mathcal{L}_2(\boldsymbol{w},\boldsymbol{\delta},\tilde{\boldsymbol{y}}) = \boldsymbol{w}^T(\boldsymbol{\Delta\phi}(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{y}}) + \boldsymbol{\delta}) + \Delta(\boldsymbol{y},\tilde{\boldsymbol{y}})$. Then we will have:*

$$\sup_{\boldsymbol{\delta}\in\Delta^2\Phi(\boldsymbol{x},\boldsymbol{y}),\tilde{\boldsymbol{y}}} \mathcal{L}_2(\boldsymbol{w},\boldsymbol{\delta},\tilde{\boldsymbol{y}}) \geq \sup_{\tilde{\boldsymbol{x}}\in\mathcal{S}(\boldsymbol{x},\boldsymbol{y}),\tilde{\boldsymbol{y}}} \mathcal{L}_1(\boldsymbol{w},\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}},\boldsymbol{y})$$

Proof sketch. The left-hand side of the inequality is equal to the right-hand side except that the supremum is taken over a superset of function values. Thus, the left-hand side cannot be any less than the right-hand side. □

Now, we can rewrite the robust formulation in (6) over variations in the feature space:

$$\underset{\boldsymbol{w}}{\text{minimize}}\, Cf(\boldsymbol{w}) + \sup_{\boldsymbol{\delta}\in\Delta^2\Phi(\boldsymbol{x},\boldsymbol{y}),\tilde{\boldsymbol{y}}} \mathcal{L}(\boldsymbol{w},\boldsymbol{\delta},\tilde{\boldsymbol{y}}) \quad (8)$$

where $\mathcal{L}(\boldsymbol{w},\boldsymbol{\delta},\tilde{\boldsymbol{y}}) = \boldsymbol{w}^T(\boldsymbol{\Delta\phi}(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{y}}) + \boldsymbol{\delta}) + \Delta(\boldsymbol{y},\tilde{\boldsymbol{y}})$.

By Lemma 3.2, the objective of (8) is an upper-bound for the objective of (6); therefore, the formulation of the problem in (8) is an approximate, but more tractable, solution for (6).

In the next section, we will show that for a wide class of uncertainty sets $\Delta^2\Phi(\boldsymbol{x},\boldsymbol{y})$, problem (8) reduces to an optimization program which can be solved as efficiently as an ordinary 1-slack structural SVM.

[1] For more on finite difference approximations, refer to Smith (1985).

4. Mapping the Uncertainty Sets

In many real world problems, there exists some expert knowledge about the uncertainty sets in the sample space. For example, for the web page classification problem, a spammer can modify web pages by adding and removing words and links, but is constrained by the cost of compromising legitimate web pages, which takes time and effort, or obfuscating spam pages, which may make them less effective at gaining clicks. We can approximate this with a simple budget on the number of words and links the adversary can change over the entire dataset. Even when such information is not readily available, it may be possible to infer an uncertainty set from training data. For example, if our dataset contains outliers, we can pair each outlier ($\tilde{\boldsymbol{x}}$) with the most similar non-outlier ($\boldsymbol{x}$) and take the differences as possible directions of manipulation: $\Delta\boldsymbol{x} = \tilde{\boldsymbol{x}} - \boldsymbol{x}$. The convex hull of these difference vectors (or an approximation thereof) can be used to define an uncertainty set for any instance.

Lemma 3.2 states that the robust formulation in feature space is a reasonable approximation for the robust formulation in the sample space, but it does not suggest any mechanical procedure for calculating the uncertainty sets in feature space from the ones in the sample space. We now derive such procedures for certain types of uncertainty sets and feature functions.

Many features of interest, including logical conjunctions, can be represented as products of several variables. We define a *multinomial feature function* as a sum of many such products:

$$\phi_{\mathcal{C}}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{(c_x, c_y)\in\mathcal{C}} \prod_{i\in c_x} \boldsymbol{x}_i \prod_{i\in c_y} \boldsymbol{y}_i \tag{9}$$

where $\mathcal{C}$ is a set of variable groups and (c_x, c_y) are the index sets of the attribute and output variables in each group. Using terminology from Markov networks, we refer to each of these variable groups as a *clique*. The summation groups together many products that share the same pattern into a single, aggregate feature so that they may be considered collectively. For example, in web page classification, the multinomial feature $\sum_i x_{i,j} y_i$ could represent the number of web pages with label 1 that contain word j. This is equivalent to having many features with tied weights.

To relate uncertainty sets in sample space to uncertainty sets in feature space, we begin with the following lemma, which bounds the disturbance of a single feature.

Lemma 4.1. *If the feature function $\phi_{\mathcal{C}}(\boldsymbol{x}, \boldsymbol{y})$ is multinomial with $\boldsymbol{0} \le \boldsymbol{x}, \boldsymbol{y} \le \boldsymbol{1}$, then its disturbance $\delta^{\mathcal{C}}$ can be upper-bounded by a function of the variations in the sample space, according to the following inequality:*

$$\frac{|\delta^{\mathcal{C}}|^p}{\alpha_{\mathcal{C}}|\mathcal{C}|^{\frac{p}{q}}} \le \sum_{c_x\in\mathcal{C}} \sum_{i\in c_x} |\tilde{x}_i - x_i|^p \tag{10}$$

where $p \ge 1$ is an arbitrary power value and $\frac{1}{p} + \frac{1}{q} = 1$; $\alpha = \max_{c_x\in\mathcal{C}} |c_x|^{(p-1)}$; $|c_x|$ is the number of evidence variables in c_x; and $|\mathcal{C}|$ is the number of different sets c_x in $\mathcal{C}$.

The proof can be found in the supplementary material.

Now we show how to apply Lemma 4.1 to obtain bounds over all features simultaneously. The next theorem is the main result of this section.

Theorem 4.2. *For multinomial feature functions and spherical uncertainty sets in the sample space $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y}) = \{\tilde{\boldsymbol{x}} \mid \|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_p \le B\}$ (with $p \ge 1$), one can construct an ellipsoidal uncertainty set in the feature space:*

$$\Delta^2\Phi(\boldsymbol{x}, \boldsymbol{y}) = \{\boldsymbol{\delta} \mid \ \|\boldsymbol{M}\boldsymbol{\delta}\|_p \le 1\} \tag{11}$$

where $\boldsymbol{M}$ is a diagonal matrix with $\frac{1}{B(d\alpha_i)^{\frac{1}{p}}|\mathcal{C}_i|^{\frac{1}{q}}}$ at the (i, i)th position. d, α_i, and $|\mathcal{C}_i|$ are appropriate constants.

Proof. Assume that $\mathcal{P} = \{\mathcal{C}_1, \ldots, \mathcal{C}_L\}$ is a set of cliques that covers all variable x_i's. Note that such a set should exist; otherwise, some variables are never used in the model. For each of the cliques, we form a corresponding difference in the feature function from Eq. (7), and apply Lemma 4.1. By adding all of the resulting inequalities, we obtain:

$$\begin{aligned}
\sum_{\mathcal{C}_i\in\mathcal{P}} \frac{|\delta^{\mathcal{C}_i}|^p}{\alpha_i|\mathcal{C}_i|^{\frac{p}{q}}} &\le d \sum_{i=1}^{\dim(\boldsymbol{x})} |\tilde{x}_i - x_i|^p \\
&= d\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_p^p \le dB^p \\
\Rightarrow \quad \sum_{\mathcal{C}_i\in\mathcal{P}} & \frac{|\delta^{\mathcal{C}_i}|^p}{B^p d\alpha_i|\mathcal{C}_i|^{\frac{p}{q}}} \le 1 \\
\Rightarrow \quad \sum_{\mathcal{C}_i\in\mathcal{P}} & \left(\frac{1}{B(d\alpha_i)^{\frac{1}{p}}|\mathcal{C}_i|^{\frac{1}{q}}}|\delta^{\mathcal{C}_i}|\right)^p \le 1
\end{aligned}$$

where $\alpha_i = \max_{c_x\in\mathcal{C}_i} |c_x|^{(p-1)}$, and $|c_x|$ is the number of variables in c_x. Since it is possible that cliques cover overlapping sets of variables, the coefficient $d \ge 1$ will be used to maintain the inequality.

Now let $\frac{1}{B(d\alpha_i)^{\frac{1}{p}}|\mathcal{C}_i|^{\frac{1}{q}}}$ be the diagonal entry in matrix $\boldsymbol{M}$ that corresponds to feature disturbance $\delta^{\mathcal{C}_i}$. For this choice of $\boldsymbol{M}$, $\|\boldsymbol{M}\boldsymbol{\delta}\|_p \le 1$. □

We have an example of applying Theorem 4.2 in Section 6.2, which will show how this construction works in practice.

Corollary 4.3. *If $\mathcal{S}(\boldsymbol{x}, \boldsymbol{y}) = \{\tilde{\boldsymbol{x}} \mid \|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_1 \leq B\}$, then $\boldsymbol{M}$ can be constructed by setting $\frac{1}{Bd}$ as its (i, i)th element, which results in a tighter upper bound.*

The proof can be found in the supplementary material.

5. Robust Optimization Programs

Our main contribution in this paper is achieving robust formulations that can be efficiently solved. We do this by demonstrating a connection between robustness to certain perturbations in feature space and certain types of weight regularization. In this section we derive formulations for achieving robust weight learning in structural SVMs when $\Delta^2\Phi(\boldsymbol{x}, \boldsymbol{y})$ is an ellipsoid, a polyhedron, or the intersection of an ellipsoid and a polyhedron.

5.1. Ellipsoidal Constrained Uncertainty

We first consider the case when the uncertainty set $\Delta^2\Phi(\boldsymbol{x}, \boldsymbol{y})$ is ellipsoidal. Recall that any ellipsoid can be represented in the form of $\{\boldsymbol{t} \mid \|\boldsymbol{M}\boldsymbol{t}\| \leq 1\}$, where $\|.\|$ is the relevant norm.

Theorem 5.1. *For $\Delta^2\Phi(\boldsymbol{x}, \boldsymbol{y}) = \{\boldsymbol{\delta} \mid \|\boldsymbol{M}\boldsymbol{\delta}\| \leq 1\}$ where $\boldsymbol{M}$ is positive definite, the optimization program of the robust structural SVM in (8) reduces to the following regularized formulation of the ordinary 1-slack structural SVM:*

$$\begin{aligned} &\underset{\boldsymbol{w},\zeta}{\text{minimize}}\; Cf(\boldsymbol{w}) + \|\boldsymbol{M}^{-1}\boldsymbol{w}\|^* + \zeta \\ &\text{subject to} \\ &\qquad \zeta \geq \sup_{\tilde{\boldsymbol{y}}} \boldsymbol{w}^T \boldsymbol{\Delta\phi}(\boldsymbol{x}, \boldsymbol{y}, \tilde{\boldsymbol{y}}) + \Delta(\boldsymbol{y}, \tilde{\boldsymbol{y}}) \end{aligned} \tag{12}$$

where $\|.\|^*$ is the dual norm of $\|.\|$.

Proof. We begin with the robust formulation of a structural SVM from (8), where the uncertainty set of $\boldsymbol{\delta}$ is defined by the ellipsoid $\|\boldsymbol{M}\boldsymbol{\delta}\| \leq 1$:

$$\underset{\boldsymbol{w}}{\text{minimize}}\; Cf(\boldsymbol{w}) + \sup_{\|\boldsymbol{M}\boldsymbol{\delta}\| \leq 1, \tilde{\boldsymbol{y}}} \mathcal{L}(\boldsymbol{w}, \boldsymbol{\delta}, \tilde{\boldsymbol{y}})$$

Let $\boldsymbol{\nu} = \boldsymbol{M}\boldsymbol{\delta}$, so that $\boldsymbol{\delta} = \boldsymbol{M}^{-1}\boldsymbol{\nu}$. Then we will have:

$$\begin{aligned} &\sup_{\|\boldsymbol{M}\boldsymbol{\delta}\| \leq 1, \tilde{\boldsymbol{y}}} \mathcal{L}(\boldsymbol{w}, \boldsymbol{\delta}, \tilde{\boldsymbol{y}}) \\ =\; &\sup_{\|\boldsymbol{M}\boldsymbol{\delta}\| \leq 1, \tilde{\boldsymbol{y}}} \boldsymbol{w}^T(\boldsymbol{\Delta\phi}(\boldsymbol{x}, \boldsymbol{y}, \tilde{\boldsymbol{y}}) + \boldsymbol{\delta}) + \Delta(\boldsymbol{y}, \tilde{\boldsymbol{y}}) \\ =\; &\sup_{\|\boldsymbol{M}\boldsymbol{\delta}\| \leq 1} \boldsymbol{w}^T\boldsymbol{\delta} + \sup_{\tilde{\boldsymbol{y}}} \boldsymbol{w}^T \boldsymbol{\Delta\phi}(\boldsymbol{x}, \boldsymbol{y}, \tilde{\boldsymbol{y}}) + \Delta(\boldsymbol{y}, \tilde{\boldsymbol{y}}) \\ =\; &\sup_{\|\boldsymbol{\nu}\| \leq 1} \boldsymbol{w}^T\boldsymbol{M}^{-1}\boldsymbol{\nu} + \sup_{\tilde{\boldsymbol{y}}} \boldsymbol{w}^T \boldsymbol{\Delta\phi}(\boldsymbol{x}, \boldsymbol{y}, \tilde{\boldsymbol{y}}) + \Delta(\boldsymbol{y}, \tilde{\boldsymbol{y}}) \end{aligned}$$

By definition of the dual norm, $\sup_{\|\boldsymbol{\nu}\| \leq 1}(\boldsymbol{w}^T\boldsymbol{M}^{-1})\boldsymbol{\nu} = \|\boldsymbol{M}^{-T}\boldsymbol{w}\|^*$. Since $\boldsymbol{M}^{-1}$ is also a definite matrix, it is symmetric; therefore, $\|\boldsymbol{M}^{-T}\boldsymbol{w}\|^* = \|\boldsymbol{M}^{-1}\boldsymbol{w}\|^*$.

$$= \quad \|\boldsymbol{M}^{-1}\boldsymbol{w}\|^* + \sup_{\tilde{\boldsymbol{y}}} \boldsymbol{w}^T \boldsymbol{\Delta\phi}(\boldsymbol{x}, \boldsymbol{y}, \tilde{\boldsymbol{y}}) + \Delta(\boldsymbol{y}, \tilde{\boldsymbol{y}})$$

By substitution, the rest of the proof is straightforward. □

Note that Theorem 5.1 can still be applied when $\boldsymbol{M}$ is not positive definite by using the Moore-Penrose inverse of $\boldsymbol{M}$ instead of the regular inverse. The result in Theorem 5.1 uses the technique of robust linear programming with arbitrary norms that is introduced in Bertsimas et al. (2004). This theorem can also be seen as a generalization of Theorem 3 in Xu et al. (2009) to structural SVMs. Theorem 5.1 shows the direct connection between the robust formulation and regularization of the non-robust formulation for structural SVMs.

Corollary 5.2. *For disturbances of the form $\|\boldsymbol{\delta}\| \leq B$ in the feature space, with B being a maximum budget for the applicable changes and $\|.\|$ being an arbitrary norm, robustness can be achieved by adding the regularization function $B\|\boldsymbol{w}\|^*$ to the objective.*

Proof. Since $\|\boldsymbol{\delta}\|/B \leq 1 \Rightarrow \|\frac{1}{B}\boldsymbol{\delta}\| = \|\frac{1}{B}\boldsymbol{I}\boldsymbol{\delta}\| \leq 1$. Let $\boldsymbol{M} = \frac{1}{B}\boldsymbol{I}$, then $\boldsymbol{M^{-1}} = B\boldsymbol{I}$. Thus, $\|\boldsymbol{M}^{-1}\boldsymbol{w}\|^* = \|B\boldsymbol{I}\boldsymbol{w}\|^* = B\|\boldsymbol{w}\|^*$. By Theorem 5.1, $B\|\boldsymbol{w}\|^*$ is the appropriate regularization function. □

Note that $\boldsymbol{M}$ can also be seen as a tuning parameter. In particular, if there is a low-dimensional representation of $\boldsymbol{M}$, then tuning $\boldsymbol{M}$ might be an option.

The commonly used L_2 regularization can be in fact interpreted as a regularization function that enforces robustness to disturbances in the feature space that are restricted to a hypersphere.

Corollary 5.3. *If $f(\boldsymbol{w}) = 0$, then setting $\boldsymbol{M} = \frac{1}{C}I$ and $\|.\| = \|.\|_2$ will recover the commonly used L_2-regularized structural SVM.*

Proof. If $\boldsymbol{M} = \frac{1}{C}I$, then $\boldsymbol{M^{-1}} = CI$. Note that the L_2 norm is dual to itself. Therefore, $f(\boldsymbol{w}) + \|\boldsymbol{M}^{-1}\boldsymbol{w}\|_2^* = 0 + \|CI\boldsymbol{w}\|_2 = C\|\boldsymbol{w}\|_2$. □

Corollary 5.4. *Robustness to variations restricted by a Mahalanobis norm $\|\boldsymbol{\delta}\|_{\boldsymbol{S}} = \sqrt{\boldsymbol{\delta}^T\boldsymbol{S}\boldsymbol{\delta}} \leq 1$, where $\boldsymbol{S}$ is positive definite, is equivalent to adding the regularization function $\|\boldsymbol{w}\|_{\boldsymbol{S}^{-1}} = \sqrt{\boldsymbol{w}^T\boldsymbol{S}^{-1}\boldsymbol{w}}$ to the objective.*

Proof. Let $\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$ be the spectral decomposition of $\boldsymbol{S}$. Set $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{U}^T$ and the norm $\|.\|$ to $\|.\|_2$. Then $\|\boldsymbol{M}\boldsymbol{\delta}\|_2 = \sqrt{\boldsymbol{\delta}^T\boldsymbol{M}^T\boldsymbol{M}\boldsymbol{\delta}} = \sqrt{\boldsymbol{\delta}^T\boldsymbol{M}^2\boldsymbol{\delta}} = \sqrt{\boldsymbol{\delta}^T\boldsymbol{S}\boldsymbol{\delta}}$. Therefore the resulting regularization function will be $\|\boldsymbol{M}^{-1}\boldsymbol{w}\|_2^* = \|\boldsymbol{M}^{-1}\boldsymbol{w}\|_2 = \sqrt{\boldsymbol{w}^T\boldsymbol{M}^{-T}\boldsymbol{M}^{-1}\boldsymbol{w}} =$

$\sqrt{\boldsymbol{w}^T\boldsymbol{U}\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{U}^T\boldsymbol{w}} = \sqrt{\boldsymbol{w}^T\boldsymbol{U}\boldsymbol{\Lambda}^{-1}\boldsymbol{U}^T\boldsymbol{w}} = \sqrt{\boldsymbol{w}^T\boldsymbol{S}^{-1}\boldsymbol{w}} = \|\boldsymbol{w}\|_{\boldsymbol{S}^{-1}}$, Note that $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$ because $\boldsymbol{U}$ is a unitary matrix. □

5.2. Polyhedral Constrained Uncertainty

For some problems, an ellipsoid may not be a good representation of the uncertainty set, but almost any convex uncertainty set can be approximated by a polyhedron. In this subsection we consider the situations in which we are aware of the shape of the polyhedral constraints on the variations in the feature space; i.e., $\Delta^2\Phi(\boldsymbol{x},\boldsymbol{y}) = \{\boldsymbol{\delta}|\boldsymbol{A}\boldsymbol{\delta} \leq \boldsymbol{b}\}$. The next theorem shows that polyhedral uncertainty sets are equivalent to linear regularization in a transformed feature space. We begin with a supporting lemma.

Lemma 5.5. *If $\boldsymbol{x} \in \mathcal{S}(\boldsymbol{x},\boldsymbol{y})$, then for the corresponding $\Delta^2\Phi(\boldsymbol{x},\boldsymbol{y}) = \{\boldsymbol{\delta}|\boldsymbol{A}\boldsymbol{\delta} \leq \boldsymbol{b}\}$, $\boldsymbol{b}$ is a non-negative vector.*

Proof. $\boldsymbol{x} \in \mathcal{S}(\boldsymbol{x},\boldsymbol{y})$, and $\boldsymbol{\phi}(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}) - \boldsymbol{\phi}(\tilde{\boldsymbol{x}},\boldsymbol{y}) = \boldsymbol{\phi}(\boldsymbol{x},\tilde{\boldsymbol{y}}) - \boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{y}) + \boldsymbol{\delta}$. Therefore, when $\tilde{\boldsymbol{x}} = \boldsymbol{x}$ then $\boldsymbol{\delta} = \boldsymbol{0}$, so we should have $\boldsymbol{0} \in \Delta^2\Phi(\boldsymbol{x},\boldsymbol{y})$. Therefore, for $\boldsymbol{\delta} = \boldsymbol{0}$, $\boldsymbol{A}\boldsymbol{\delta} = \boldsymbol{A}\boldsymbol{0} \leq \boldsymbol{b}$; i.e., $\boldsymbol{b} \geq \boldsymbol{0}$. □

Theorem 5.6. *For $\Delta^2\Phi(\boldsymbol{x},\boldsymbol{y}) = \{\boldsymbol{\delta}|\boldsymbol{A}\boldsymbol{\delta} \leq \boldsymbol{b}\}$, the optimization program of the robust structural SVM in (8) reduces to the following ordinary 1-slack structural SVM*

$$\begin{aligned} &\underset{\boldsymbol{\lambda}\geq 0,\zeta}{\text{minimize}}\ \ Cf(\boldsymbol{A}^T\boldsymbol{\lambda}) + \boldsymbol{\lambda}^T\boldsymbol{b} + \zeta \qquad (13)\\ &\text{subject to}\ \ \zeta \geq \sup_{\tilde{\boldsymbol{y}}} \boldsymbol{\lambda}^T\boldsymbol{A}\boldsymbol{\Delta\phi}(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{y}}) + \Delta(\boldsymbol{y},\tilde{\boldsymbol{y}}) \end{aligned}$$

Proof. By substituting the uncertainty set $\Delta^2\Phi(\boldsymbol{x},\boldsymbol{y}) = \{\boldsymbol{\delta}|\boldsymbol{A}\boldsymbol{\delta} \leq \boldsymbol{b}\}$ into the optimization program (8), we obtain:

$$\underset{\boldsymbol{w}\geq 0}{\text{minimize}}\ \ Cf(\boldsymbol{w}) + \sup_{\boldsymbol{A\delta}\leq\boldsymbol{b},\tilde{\boldsymbol{y}}} \mathcal{L}(\boldsymbol{w},\boldsymbol{\delta},\tilde{\boldsymbol{y}}) \qquad (14)$$

We can rewrite $\sup_{\boldsymbol{A\delta}\leq\boldsymbol{b},\tilde{\boldsymbol{y}}} \mathcal{L}(\boldsymbol{w},\boldsymbol{\delta},\tilde{\boldsymbol{y}})$ as:

$$\begin{aligned} &\sup_{\boldsymbol{A\delta}\leq\boldsymbol{b},\tilde{\boldsymbol{y}}} \boldsymbol{w}^T(\boldsymbol{\Delta\phi}(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{y}}) + \boldsymbol{\delta}) + \Delta(\boldsymbol{y},\tilde{\boldsymbol{y}})\\ =\ &\sup_{\boldsymbol{A\delta}\leq\boldsymbol{b}} \boldsymbol{w}^T\boldsymbol{\delta} + \sup_{\tilde{\boldsymbol{y}}} \boldsymbol{w}^T\boldsymbol{\Delta\phi}(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{y}}) + \Delta(\boldsymbol{y},\tilde{\boldsymbol{y}}) \end{aligned}$$

We perform a Lagrangian relaxation on $\boldsymbol{A}\boldsymbol{\delta} \leq \boldsymbol{b}$:

$$\begin{aligned} =\ &\inf_{\boldsymbol{\lambda}\geq 0}\sup_{\boldsymbol{\delta}}(\boldsymbol{w}^T\boldsymbol{\delta} - \boldsymbol{\lambda}^T\boldsymbol{A}\boldsymbol{\delta} + \boldsymbol{\lambda}^T\boldsymbol{b})\\ &+ \sup_{\tilde{\boldsymbol{y}}} \boldsymbol{w}^T\boldsymbol{\Delta\phi}(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{y}}) + \Delta(\boldsymbol{y},\tilde{\boldsymbol{y}})\\ =\ &\inf_{\boldsymbol{\lambda}\geq 0}\left(\boldsymbol{\lambda}^T\boldsymbol{b} + \sup_{\boldsymbol{\delta}}(\boldsymbol{w}^T - \boldsymbol{\lambda}^T\boldsymbol{A})\boldsymbol{\delta}\right)\\ &+ \sup_{\tilde{\boldsymbol{y}}} \boldsymbol{w}^T\boldsymbol{\Delta\phi}(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{y}}) + \Delta(\boldsymbol{y},\tilde{\boldsymbol{y}}) \end{aligned}$$

Note that the value of the $\sup_{\boldsymbol{\delta}}(\boldsymbol{w}^T - \boldsymbol{\lambda}^T\boldsymbol{A})\boldsymbol{\delta}$ will be $+\infty$, unless $\boldsymbol{w} = \boldsymbol{A}^T\boldsymbol{\lambda}$, therefore:

$$= \begin{cases} \inf_{\boldsymbol{\lambda}\geq 0}\boldsymbol{\lambda}^T\boldsymbol{b} + \sup_{\tilde{\boldsymbol{y}}}\,[\boldsymbol{w}^T\boldsymbol{\Delta\phi}(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{y}}) + \Delta(\boldsymbol{y},\tilde{\boldsymbol{y}})] & \\ & \text{if } \boldsymbol{w} = \boldsymbol{A}^T\boldsymbol{\lambda}\\ +\infty & \text{otherwise.} \end{cases}$$

Therefore (14) can be rewritten as:

$$\begin{aligned} &\underset{\boldsymbol{w}\geq 0}{\text{minimize}}\ Cf(\boldsymbol{w}) + \inf_{\boldsymbol{\lambda}\geq 0}\boldsymbol{\lambda}^T\boldsymbol{b} +\\ &\qquad \sup_{\tilde{\boldsymbol{y}}} \boldsymbol{w}^T\boldsymbol{\Delta\phi}(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{y}}) + \Delta(\boldsymbol{y},\tilde{\boldsymbol{y}})\\ &\text{subject to}\quad \boldsymbol{w} = \boldsymbol{A}^T\boldsymbol{\lambda} \qquad (15) \end{aligned}$$

By substituting $\boldsymbol{w}$ with $\boldsymbol{A}^T\boldsymbol{\lambda}$, (15) can be equivalently written as (13). Note that by Lemma (5.5), the value of $\boldsymbol{b}$ is is always non-negative, so no value of $\boldsymbol{\lambda}$ can lead the value of the objective in the outer minimization to negative infinity. □

It is a known fact that maximization (or minimization) of L_1 and L_∞ norms of affine functions can be converted to linear programs (Boyd & Vandenberghe, 2004). In the following proposition, we state that both Theorem 5.1 and Theorem 5.6 will lead to equivalent optimization programs in these cases.

Proposition 5.7. *If the disturbances in the feature space are restricted by some ellipsoid that is defined by L_1 or L_∞ norms, then the optimization program that is generated by Theorem 5.1 can be equivalently transformed to one that is generated by Theorem 5.6*

The proof can be found in the supplementary materials.

5.3. Ellipsoidal/Polyhedral Conjunction

In some cases, the uncertainty set in feature space may resemble an ellipsoid but with additional linear constraints. We can model this as the intersection of an ellipsoid and a polyhedron. The following theorem describes how such uncertainty sets can be transformed into regularizers.

Theorem 5.8. *For $\Delta^2\Phi(\boldsymbol{x},\boldsymbol{y}) = \{\boldsymbol{\delta}|\|\boldsymbol{M}\boldsymbol{\delta}\| \leq 1, \boldsymbol{A}\boldsymbol{\delta} \leq \boldsymbol{b}\}$, the optimization program of the robust structural SVM in (8) reduces to the following ordinary 1-slack structural SVM:*

$$\begin{aligned} &\underset{\boldsymbol{w},\boldsymbol{\lambda}\geq 0,\zeta}{\text{minimize}}\ Cf(\boldsymbol{w}) + \|\boldsymbol{M}^{-1}(\boldsymbol{w} - \boldsymbol{A}^T\boldsymbol{\lambda})\|^* + \boldsymbol{b}^T\boldsymbol{\lambda} + \zeta\\ &\text{subject to}\\ &\qquad \zeta \geq \sup_{\tilde{\boldsymbol{y}}} \boldsymbol{w}^T\boldsymbol{\Delta\phi}(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{y}}) + \Delta(\boldsymbol{y},\tilde{\boldsymbol{y}}) \qquad (16) \end{aligned}$$

The proof of Theorem 5.8 is a combination of the proofs of Theorems 5.1 and 5.6. First, we perform the Lagrangian relaxation as in the proof of 5.6, and then we add the dual of $\boldsymbol{M}^{-1}(\boldsymbol{w}-\boldsymbol{A}^T\boldsymbol{\lambda})$ (the coefficient of $\boldsymbol{\delta}$) as the regularization term.

The results in Theorems 5.1, 5.6, and 5.8 apply to binary and multi-class SVMs as well simply by restricting the space of y to a small set of values. For Theorem 5.1, this reduces to results proved by Xu et al. (2009). For the later theorems, we are not aware of any analogous previous work for binary or multi-class SVMs.

Some limiting cases of Theorem 5.8 are also interesting. For example, for a (geometrically) infinitely large polyhedron $\boldsymbol{A\delta} \leq \boldsymbol{b}$ (e.g., elements of the vector $\boldsymbol{b}$ are infinitely large), $\boldsymbol{\lambda}$ must be $\boldsymbol{0}$, which recovers the regularization term $\|\boldsymbol{M}^{-1}\boldsymbol{w}\|^*$ introduced in Theorem 5.1.

Let $\lambda_1, \ldots, \lambda_m$ be the eigenvalues of $\boldsymbol{M}$. If $\min(\lambda_i) \to +\infty$ (for example, a diagonal matrix with very large numbers on the diagonal), then as a result $\boldsymbol{\delta} \to \boldsymbol{0}$ in the robust formulation. Intuitively, this means that the uncertainty set only contains the unmodified input $\boldsymbol{x}$. In this case, $\boldsymbol{M}^{-1}$ approaches the zero matrix, and as a result the regularization term $\|\boldsymbol{M}^{-1}(\boldsymbol{w} - \boldsymbol{A}^T\boldsymbol{\lambda})\|^*$ fades as expected. On the other hand, if $\max(\lambda_i) \to 0$, then $\|\boldsymbol{M}^{-1}(\boldsymbol{w} - \boldsymbol{A}^T\boldsymbol{\lambda})\|^* \approx \|L_{\boldsymbol{M}}\boldsymbol{I}(\boldsymbol{w}-\boldsymbol{A}^T\boldsymbol{\lambda})\|^* = L_{\boldsymbol{M}}\|(\boldsymbol{w}-\boldsymbol{A}^T\boldsymbol{\lambda})\|^*$, where $L_{\boldsymbol{M}} \to +\infty$. Therefore, the constraint $\boldsymbol{w} = \boldsymbol{A}^T\boldsymbol{\lambda}$ must be satisfied, leading to (13).

6. Experiments

We demonstrate the utility of our approach by applying it to a collective classification problem.

6.1. Dataset

We introduce a new dataset based on the political blogs dataset collected by Adamic and Glance (2005). The original dataset consists of 1490 blogs and their network structure from the 2004 presidential election period. Each blog is labeled as liberal or conservative. We expanded this dataset by crawling the actual blog texts in different years to obtain a vector of 250 word features for each blog in each yearly snapshot from 2003 to 2013. We used the internet archive website (https://archive.org/web/) to obtain snapshots of each blog in each year. We selected the snapshot closest to October 10th of each year and removed blogs that were inactive for an 8 month window (4 months before and after October 10th).

The political affiliation of a blog can thus be inferred from both the words on the blog and its hyperlink relationships to other blogs, which are likely to have similar political views. Since political topics evolve quickly over time, we expect a significant amount of concept drift over the years, especially over the word features. Since the test distribution is evolving significantly, we might expect a robust model to outperform a non-robust model when trained and tested on different years.[2]

6.2. Problem Formulation

In our experiments, we use both word features and link features. We construct one multinomial feature for each word i and label k, $\phi_{ik}(\boldsymbol{x}, \boldsymbol{y}) = \sum_j x^w_{ji} y_{jk}$, where $x^w_{ji} = 1$ if the jth blog contains the ith word, and $y_{jk} = 1$ if the jth blog has label k. We also construct a link feature for each label k: $\phi_k(\boldsymbol{x}, \boldsymbol{y}) = \sum_{ij} x^e_{ij} y_{ik} y_{jk}$, where $x^e_{ij} = 1$ if there is a link from the ith blog to the jth blog.

For our constraints, we assume that the number of words added or removed is bounded by some budget, B_w, and the number of edges by another budget, B_e. Thus, letting $\boldsymbol{x}^w$ be vector of all word-related variables, $\|\tilde{\boldsymbol{x}}^w - \boldsymbol{x}^w\|_1 \leq B_w$. Similarly, $\|\tilde{\boldsymbol{x}}^e - \boldsymbol{x}^e\|_1 \leq B_e$.

In order to construct the uncertainty set in the feature space, we follow the construction procedure in Theorem 4.2 and then apply Corollary 4.3. For the word features ϕ_{ik} and edge features ϕ_k we can construct separate uncertainty sets:

$$\begin{aligned}
|\delta_{ik}| &\leq \sum_i |\tilde{x}^w_{ik} - x^w_{ik}| \\
\Rightarrow \sum_k |\delta_{ik}| &\leq \sum_{i,k} |\tilde{x}^w_{ik} - x^w_{ik}| = \|\tilde{\boldsymbol{x}}^w - \boldsymbol{x}^w\| \leq B_w \\
|\delta_{ek}| &\leq \sum_{i,j} |\tilde{x}^e_{ij} - x^e_{ij}| = \|\tilde{\boldsymbol{x}}^e - \boldsymbol{x}^e\| \leq B_e
\end{aligned}$$

In our domain there are two classes, liberal and conservative, so $k \in \{0, 1\}$. As a result: $\sum_{k=0}^{1} \sum_i \frac{|\delta_{ik}|}{2B_w} \leq 1$, and $\sum_{k=0}^{1} \frac{|\delta_{ek}|}{2B_e} \leq 1$. Summing the equalities and dividing by two:

$$\sum_{i,k} \frac{|\delta_{ik}|}{4B_w} + \sum_k \frac{|\delta_{ek}|}{4B_e} \quad \leq \quad 1$$

Finally, let $\boldsymbol{\delta} = [\delta_{11}, \ldots, \delta_{nm}, \delta_{e0}, \delta_{e1}]^T$, where $m = 250$ is the number of word attributes that are chosen from training data, and n is the number of the nodes in the graph. Then, $\boldsymbol{M}$ is a diagonal matrix with entries $[\frac{1}{4B_w}, \ldots, \frac{1}{4B_w}, \frac{1}{4B_e}, \frac{1}{4B_e}]$, so we will have $\|\boldsymbol{M\delta}\|_1 \leq 1$. Note that, in this uncertainty translation, the base case of Lemma 4.1 holds in the first place, so the inequality is in its tightest form.

[2] The expanded political blogs dataset and our robust SVM implementation can be downloaded from the following URL: http://ix.cs.uoregon.edu/~lowd/robustsvmstruct.

6.3. Methods and Results

We partitioned the blogs into three separate sub-networks and used three-way cross-validation, training on one subnetwork, using the next as a validation set for tuning parameters, and evaluating on the third. We used mutual information to select the 250 most informative words separately for each training set. However, rather than training, tuning, and testing on the same year, we trained and tuned on the snapshot from 2004 and evaluated the models on every snapshot from 2003 to 2013.

Standard structural SVMs have one parameter C that needs to be tuned. The robust method has an additional regularization parameter $C' = 1/B_e = 1/B_w$ which scales the strength of the robust regularization.[3] We chose these parameters from the semi-logarithmic set $\{0, .001, .002, .005, .1, \ldots, 10, 20, 50\}$. We intentionally added 0 to this set to allow removing one of the regularization terms. We learned parameters using a cutting plane method, implemented using the Gurobi optimization engine 5.60 (2014) for running all integer and quadratic programs. We ran for 50 iterations and selected the weights from the iteration with the best performance on the tuning set.

Figure 1 shows the average error rate of the robust and non-robust formulations in each year. In 2004, both have very similar accuracy. This is not surprising, since they were tuned for this particular year. In years before and after 2004, the error rate increases for both models. However, the error rate of the robust model is often substantially lower than the non-robust model. We attribute this to the fact that the robust model has additional L_∞ regularization (since L_∞ is the dual of the L_1 uncertainty set used). This prevents the model from relying too much on a small set of features that may change, such as a particular political buzzword that might go out of fashion. These results demonstrate that robust methods for learning structural SVMs can lead to large improvements in accuracy, even when we do not have an explicit adversary or a perfect model of the perturbations.

7. Related Work

In this paper, the big picture of our formulation for robustness in the presented algorithms is based on a minimax formulation, where the learner minimizes a loss function and, at the same time, the antagonistic adversary tries to maximize the same quantity. Some related work has focused on designing classifiers that are robust to adversarial perturbation of the input data in a minimax formulation. For example, Globerson and Roweis (2006) introduce a classifier that is robust to feature deletion. Teo et al. (2008) extend this to any adversarial manipulation that can be efficiently simulated. Livni et al. (2012) show that a minimax formulation of robustness in the presence of stochastic adversaries results in L_2 (Frobenius for matrix weights) regularization, and for the multi-class case results in two-infinity regularization of the model weights. Torkamani and Lowd (2013), show that for associative Markov networks, robust weight learning for collective classification can be efficiently done with a convex quadratic program.

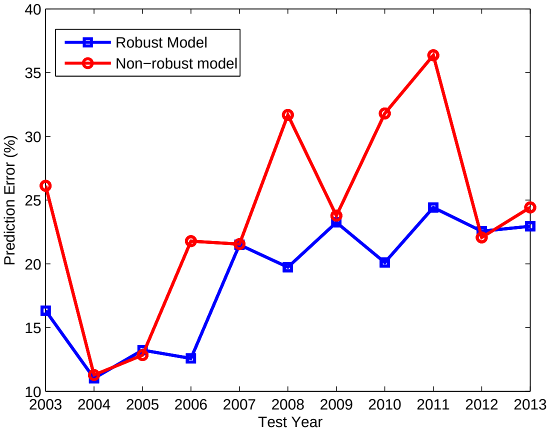

Figure 1. Average prediction error of robust and non-robust models trained on year 2004 and evaluated on years 2003-2013.

Xu et al.'s work on robustness and regularization (2009) is the most related previous work, which analyzes the connection between robustness and regularization in binary SVMs. Our work goes well beyond these results (and the ones mentioned in the introduction) by analyzing arbitrary structural SVMs and showing how they can be made robust without directly simulating the adversary, by choosing the appropriate regularization function.

Acknowledgments

We thank the anonymous reviewers for useful comments. This research was partly funded by ARO grant W911NF-08-1-0242 and NSF grant OCI-0960354. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ARO, NSF, or the U.S. Government.

References

Adamic, L.A. and Glance, N. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd International Workshop on Link Discovery*, pp. 36–43. ACM, 2005.

[3] In general, B_e and B_w could be tuned separately, but we did not do this in our experiments.

Ben-Tal, A. and Nemirovski, A. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

Ben-Tal, A. and Nemirovski, A. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25 (1):1–13, 1999.

Ben-Tal, A. and Nemirovski, A. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88(3):411–424, 2000.

Ben-Tal, A. and Nemirovski, A. On polyhedral approximations of the second-order cone. *Mathematics of Operations Research*, 26(2):193–205, 2001.

Bertsimas, D. and Sim, M. The price of robustness. *Operations research*, 52(1):35–53, 2004.

Bertsimas, D., Pachamanova, D., and Sim, M. Robust linear optimization under general norms. *Operations Research Letters*, 32(6):510–516, 2004.

Bhattacharyya, C., Pannagadatta, KS, and Smola, A. A second order cone programming formulation for classifying missing data. *Advances in Neural Information Processing Systems*, 17:153–160, 2004.

Boyd, S. and Vandenberghe, L. *Convex optimization*. Cambridge University Press, 2004.

El Ghaoui, L., Lanckriet, G.R.G., and Natsoulis, G. *Robust classification with interval data*. Computer Science Division, University of California, 2003.

Globerson, A. and Roweis, S. Nightmare at test time: robust learning by feature deletion. In *Proceedings of the Twenty-Third International Conference on Machine Learning*, pp. 353–360, Pittsburgh, PA, 2006. ACM Press.

Gurobi Optimization, Inc. Gurobi optimizer reference manual, 2014. URL http://www.gurobi.com.

Joachims, T., Finley, T., and Yu, C.-N. J. Cutting-plane training of structural SVMs. *Machine Learning*, 77(1): 27–59, 2009.

Lanckriet, G.R.G., Ghaoui, L.E., Bhattacharyya, C., and Jordan, M.I. A robust minimax approach to classification. *The Journal of Machine Learning Research*, 3: 555–582, 2003.

Livni, R., Crammer, K., and Globerson, A. A simple geometric interpretation of svm using stochastic adversaries. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 22, pp. 722–730, 2012.

Shivaswamy, Pannagadatta K, Bhattacharyya, C., and Smola, A. Second order cone programming approaches for handling missing and uncertain data. *The Journal of Machine Learning Research*, 7:1283–1314, 2006.

Smith, G. *Numerical solution of partial differential equations: finite difference methods*. Oxford University Press, 1985.

Teo, C.H., Globerson, A., Roweis, S., and Smola, A. Convex learning with invariances. In *Advances in Neural Information Processing Systems 21*, 2008.

Torkamani, M. and Lowd, D. Convex adversarial collective classification. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 642–650, 2013.

Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pp. 104. ACM, 2004.

Xu, H., Caramanis, C., and Mannor, S. Robustness and regularization of support vector machines. *The Journal of Machine Learning Research*, 10:1485–1510, 2009.