On Robustness and Regularization of Structural Support Vector Machines

MohamadAli Torkamani, Daniel Lowd
Computer and Information Science Department
University of Oregon
Eugene, OR 97403
{ali,lowd}@cs.uoregon.edu

Abstract

Previous analysis of binary SVMs has demonstrated a deep connection between robustness to perturbations over symmetric 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 where there are perturbations in the feature space. We consider two different classes of uncertainty sets for the perturbations: ellipsoidal uncertainty sets and polyhedral uncertainty sets. In both cases, we show that the robust optimization problem is equivalent to the non-robust formulation with an additional regularizer. For the ellipsoidal uncertainty set, the additional regularizer is based on the dual norm. For the polyhedral uncertainty set, we show that 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 similar results can be obtained for a combination of these two uncertainty sets.

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 data is noisy and perturbed; its distribution is constantly changing; and in some cases, such as spam filtering and fraud detection, an adversary may be actively manipulating it to defeat the learned model. In such cases, 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 perturbations are constrained to some domain-specific uncertainty set.

In this work we assume that the possible uncertain perturbations are restricted, this makes our work related to the following: Bertsimas et al. [1] show that for box-bounded disturbances in the known parameters of a linear program, there is a trade-off between optimality and robustness of the solution; they propose a robust linear programing method, for decreasing the "price of robustness" in order to prevent the model parameters to be too conservative. Later, they show that if the disturbances of the inputs are restricted to an ellipsoid around the true values defined by some norm, then the robust linear problems can be reduced to convex cone programs; where the conic constraint is defined by the dual of the original norm [2]. Recently, Xu et al. [3] analyzed the robustness of binary SVMs, where the disturbances of the input data are restricted to an ellipsoid defined by a norm; they showed that the robust formulation is equivalent to regularization of support vector machines by the dual norm.

In this paper, our goal is to achieve a structural SVM model which is robust in presence of adversarial perturbation of the test data. We focus on the robustness of structural SVMs, based on a minimax formulation. We begin with formulating the problem of learning robust structural SVMs where there are perturbations in the feature space. When these perturbations are constrained to an ellipsoid defined by some norm, we show that the robust optimization problem is equivalent to the non-robust formulation with an additional regularizer based on the dual norm. When these perturbations are constrained to a polyhedron, we show that the robust optimization problem is equivalent to adding a

linear regularizer in a transformed weight space related to the linear constraints of the polyhedron. Finally, we show that these constraint sets can be combined and demonstrate a number of interesting special cases. This represents the first robust formulation of general purpose structural support vector machines with arbitrary features functions.

Some related work has focused on designing classifiers that are robust to adversarial perturbation of the input data in similar minimax formulations. For example, Globerson et al. [4] introduce a classifier that is robust to feature deletion. Teo et al. [5] extend this to any adversarial manipulation that can be efficiently simulated. Livni et al. [6] show that a minimax formulation of robustness in the presence of stochastic adversaries results in L_2 (Frobenius for matrix weights) regularization of the model weights. Torkamani et al. [7] showed that for associative Markov networks, robust weight learning for collective classification can be efficiently done with a convex quadratic program. In this paper, we present robust formulations for general structural SVMs. In Teo et al. [4] and Torkamani et al. [7] there is an assumption that the worst-case perturbation in the instance space can be efficiently computed, which might be an intractable problem in general; in our work we do not need this assumption. Our work is also differs from Livni et al. [6], because we propose methods for general structural SVM, our formulation is derived independently of any probabilistic assumptions, and we allow different classes of possible perturbations. Among other related work that use robust optimization we can name the following: Ben-Tal and Nemirovski [8, 9, 10, 11] showed that there exist a range of applications that could be formulated in a robust convex optimization framework based on the minimax principle. A number of other authors have explored the application of robust optimization to classification problems (e.g., [12, 13, 14, 15]).

2 Structural SVMs

We consider the problem of robust learning of structured prediction models. Unlike traditional classification, which uses a small number of classes, structured prediction problems are characterized by an exponentially large space of possible outputs, such as parse trees or graph labeling. We base our analysis on structural support vector machines [16, 17, 18, 19] which are very similar to max-margin Markov networks[20, 21]. The optimization program of a 1-slack structural SVM is:

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

where $\boldsymbol{x}$ is the collection of all input variables (such as node features and edge information of a graphed dataset), $\boldsymbol{y}$ is the value of the desired structured query variables (such as a parse tree or set of the labels of nodes in a graph), $\boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{y})$ is the feature vector, and $\boldsymbol{w}$ is the vector of the model parameters. The goal is to learn $\boldsymbol{w}$; parameters are learned to maximize the margin between the true labeling $\boldsymbol{y}$ and any alternate labeling $\tilde{\boldsymbol{y}} \neq \boldsymbol{y}$. $f(\boldsymbol{w})$ is a regularization function that penalizes "large" weights. Depending on the application, $f(\boldsymbol{w})$ can be any convex function in general. We refer to the feature vector corresponding to the whole training data as $\boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{y})$, and the loss measure between $\boldsymbol{y}$ and $\tilde{\boldsymbol{y}}$ as $\Delta(\boldsymbol{y},\tilde{\boldsymbol{y}})$. For simplicity, we express this optimization problem using a single input/output pair, (x, y), but it can easily be expanded to set of N independent examples, each of which makes an independent contribution to the loss or feature functions. We refer to the value of $\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x},\tilde{\boldsymbol{y}})$ as the *score* of labeling $\boldsymbol{x}$ as $\tilde{\boldsymbol{y}}$ for the given model weights $\boldsymbol{w}$. At test time, the label that achieves the highest score value is reported as the final prediction: $\boldsymbol{y}_{prediction} = \arg\max_{\tilde{\boldsymbol{y}}} \boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x},\tilde{\boldsymbol{y}})$. For brevity, we skip other details of the structural SVM. For more details refer to [18]. For the notation, please refer to the supplementary materials file.

3 Robust optimization programs

Let $\mathcal{S}(\boldsymbol{x})$ be the set of possible (adversarial) manipulations of the input data $\boldsymbol{x}$; also let $\tilde{\boldsymbol{x}} \in \mathcal{S}(\boldsymbol{x})$ be a particular member of this set. We assume that $\boldsymbol{x} \in \mathcal{S}(\boldsymbol{x})$, which means $\boldsymbol{x}$ can remain unchanged. Let $\boldsymbol{\phi}(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}) - \boldsymbol{\phi}(\tilde{\boldsymbol{x}},\boldsymbol{y}) = \boldsymbol{\phi}(\boldsymbol{x},\tilde{\boldsymbol{y}}) - \boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{y}) + \boldsymbol{\delta}$. In other words, $\boldsymbol{\delta}$ represents how the adversary perturbs the difference between two feature vectors, the first corresponding to an alternate labeling $\tilde{\boldsymbol{y}}$ and the second corresponding to the true labeling. For a given $\boldsymbol{x}$, $\boldsymbol{y}$ and $\mathcal{S}(\boldsymbol{x})$, we can bound the set of feasible $\boldsymbol{\delta}$ values as follows: Let $\Delta^2\Phi(\boldsymbol{x},\boldsymbol{y}) \equiv \{\boldsymbol{\delta} = (\boldsymbol{\phi}(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{y}}) - \boldsymbol{\phi}(\tilde{\boldsymbol{x}},\boldsymbol{y})) - (\boldsymbol{\phi}(\boldsymbol{x},\tilde{\boldsymbol{y}}) - \boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{y}))|\ \forall\tilde{\boldsymbol{x}} \in \mathcal{S}(\boldsymbol{x}),\tilde{\boldsymbol{y}}\}$. Since $\boldsymbol{x} \in \mathcal{S}(\boldsymbol{x})$, then $\mathbf{0} \in \Delta^2\Phi(\boldsymbol{x},\boldsymbol{y})$. In the supplementary material of this paper we explain why this is a reasonable assumption. The optimization program for the robust

1-slack structural SVM will be:

$$\underset{\boldsymbol{w}}{\text{minimize}}\ Cf(\boldsymbol{w}) + \sup_{\boldsymbol{\delta}\in\Delta^2\Phi(\boldsymbol{x},\boldsymbol{y}),\tilde{\boldsymbol{y}}} \boldsymbol{w}^T(\boldsymbol{\phi}(\boldsymbol{x},\tilde{\boldsymbol{y}}) - \boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{y}) + \boldsymbol{\delta}) + \Delta(\boldsymbol{y},\tilde{\boldsymbol{y}}) \tag{2}$$

In the next subsections, we show that for a wide class of outer bounds on $\Delta^2\Phi(\boldsymbol{x},\boldsymbol{y})$'s, problem (2) reduces to an optimization program, which can be solved almost as efficiently as an ordinary 1-slack structural SVM. Our main contribution in this paper is achieving robust formulations that can be efficiently solved. Therefore, we focus on the robust formulation on feature space that we introduced in (2). In particular, we show that for cases where $\Delta^2\Phi(\boldsymbol{x},\boldsymbol{y})$ is an ellipsoid or a polyhedron, problem (2) reduces to a tractable optimization program which is very similar to ordinary 1-slack structural SVMs. We will show that these two approaches can sometimes lead to the same model, and also can be combined to encode a wider class of robust formulations. Note that, in our definition, the set of possible disturbances $\Delta^2\Phi(\boldsymbol{x},\boldsymbol{y})$ is only a function of the input, $\boldsymbol{x}$, and the labels, $\boldsymbol{y}$; while, it is independent of the alternate labeling $\tilde{\boldsymbol{y}}$.

3.1 Ellipsoidal constrained uncertainty

In this subsection, we show that if the uncertainty set $\Delta^2\Phi(\boldsymbol{x},\boldsymbol{y})$ is ellipsoidal, then the robust formulation of structural SVMs in feature space is exactly equivalent to adding extra regularization of the model weights to the objective. Recall that any ellipsoid can be re-parametrized in the form of $\{\boldsymbol{t}|\|\boldsymbol{M}\boldsymbol{t}\| \leq 1\}$, where $\|.\|$ is the relevant norm.

Theorem 3.1. *For $\Delta^2\Phi(\boldsymbol{x},\boldsymbol{y}) = \{\boldsymbol{\delta}|\|\boldsymbol{M}\boldsymbol{\delta}\| \leq 1\}$, with $\boldsymbol{M} \in \boldsymbol{S}_{++}$, the optimization program of the robust structural SVM in (2) reduces to the following regularized formulation of the ordinary 1-slack structural SVM:*

$$\begin{aligned} &\underset{\boldsymbol{w},\zeta}{\text{minimize}}\ Cf(\boldsymbol{w}) + \|\boldsymbol{M}^{-1}\boldsymbol{w}\|^* + \zeta \\ &\text{subject to}\quad \zeta \geq \sup_{\tilde{\boldsymbol{y}}} \boldsymbol{w}^T(\boldsymbol{\phi}(\boldsymbol{x},\tilde{\boldsymbol{y}}) - \boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{y})) + \Delta(\boldsymbol{y},\tilde{\boldsymbol{y}}) \end{aligned} \tag{3}$$

where $\|.\|^*$ is the dual norm of $\|.\|$. The result in Theorem 3.1 uses the technique of robust linear programming with arbitrary norms that is introduced in [2]. This theorem can also be seen as a generalization of Theorem 3 in [3] to structural SVMs. Theorem 3.1 shows the direct connection between the robust formulation of structural SVMs and regularization of the non-robust formulation for structural SVMs. To the best knowledge of the authors, this is the first such result for structural SVMs.

Corollary 3.2. *If the set of variations in feature space is in the form of $\|\boldsymbol{\delta}\| \leq B$, where B is a maximum budget for changes in the feature space and $\|.\|$ is an arbitrary norm, then robustness to these manipulations is equivalent to adding the regularization function $B\|\boldsymbol{w}\|^*$ to the objective.*

Corollary 3.3. *If $f(\boldsymbol{w}) = 0$, then setting $\boldsymbol{M} = \frac{1}{C}I$ (the diagonal matrix with $\frac{1}{C}$'s on the diagonal), and $\|.\| = \|.\|_2$, will recover the commonly used L_2-regularized structural SVM.*

Corollary 3.4. *Robustness to variations restricted by a Mahalanobis norm, i.e. $\|\boldsymbol{\delta}\|_{\boldsymbol{S}} = \sqrt{\boldsymbol{\delta}^T\boldsymbol{S}\boldsymbol{\delta}} \leq 1$, where $\boldsymbol{S} \in \boldsymbol{S}_{++}$, is equivalent to adding the regularization function $\|\boldsymbol{w}\|_{\boldsymbol{S}^{-1}} = \sqrt{\boldsymbol{w}^T\boldsymbol{S}^{-1}\boldsymbol{w}}$ to the objective.*

3.2 Polyhedral constrained uncertainty

In this section, we consider the situation when the variations are instead constrained by a polyhedron: $\Delta^2\Phi(\boldsymbol{x},\boldsymbol{y}) = \{\boldsymbol{\delta}|\boldsymbol{A}\boldsymbol{\delta} \leq \boldsymbol{b}\}$.

Theorem 3.5. *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 (2) reduces to the following ordinary 1-slack structural SVM*

$$\begin{aligned} &\underset{\boldsymbol{\lambda}\geq 0,\zeta}{\text{minimize}}\ Cf(\boldsymbol{A}^T\boldsymbol{\lambda}) + \boldsymbol{b}^T\boldsymbol{\lambda} + \zeta \\ &\text{subject to}\quad \zeta \geq \sup_{\tilde{\boldsymbol{y}}} \boldsymbol{\lambda}^T(\boldsymbol{A}\boldsymbol{\phi}(\boldsymbol{x},\tilde{\boldsymbol{y}}) - \boldsymbol{A}\boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{y})) + \Delta(\boldsymbol{y},\tilde{\boldsymbol{y}}) \end{aligned} \tag{4}$$

Theorem 3.5 states that, when the perturbations of the features are restricted to a polyhedron, then re-parameterizing the regularization function $f(.)$, applying a linear transformation to the feature

vectors, and adding a linear regularizer of the new model parameters $\boldsymbol{\lambda}$ leads to an equivalent robust formulation. Weights in the original space can be recovered as $\boldsymbol{w} = \boldsymbol{A}^T\boldsymbol{\lambda}$. We choose to express our optimization program in terms of $\boldsymbol{\lambda}$ rather than $\boldsymbol{w}$ because $\boldsymbol{A}$ is typically not invertible, so the reverse mapping does not necessarily exist.

Note that if the perturbations are underconstrained, then the solution will involve many zero weights. For example, if w_i is negative, w_j is positive, and $\delta_i + \delta_j \leq 1$ is the only constraint, then $\sup w_i\delta_i + w_j\delta_j$ can be infinitely large. Therefore the optimization will set w_i and w_j to zero. In general, if $\boldsymbol{A}$ is low-rank. i.e. $\text{rank}(\boldsymbol{A}) \neq \dim\phi$, then some of the elements of vector $\boldsymbol{\delta}$ will have no restrictions at all, which will lead the optimization program to choose the most conservative weights (e.g. $\boldsymbol{w} = \boldsymbol{0}$). Intuitively, if $\boldsymbol{A}$ is fat and short, then we are projecting $\boldsymbol{w}$ to a lower dimensional space that could be not descriptive enough. The choice of $\boldsymbol{A}$ is then important; in order to use this formulation we should have full rank $\boldsymbol{A}$, which implies that we need to have box-constraints on $\boldsymbol{\delta}$ (or full rank transformation of them; i.e., linear combinations from which we can recover box constraints) embedded into $\boldsymbol{A\delta} \leq \boldsymbol{b}$. The degenerate weights are not necessarily bad; if the disturbances really are underconstrained, then the adversary can make huge changes in the feature values so weights on those feature values are useless. This is a symptom of an overpowerful adversary, not a flawed formulation as an optimization problem. Ellipsoids defined by the L_1 and L_∞ norms can also be represented as polyhedrons. We can show that both the polyhedral and ellipsoidal robust formulations are equivalent, as expected.

3.3 Ellipsoidal/Polyhedral conjunction

In the previous two sections we developed two optimization programs for robust weight learning of structural SVMs, when the disturbance set in feature space is restricted either by a general norm or by a polyhedron. A question that arises here is the possibility of having an efficient solution when we are aware of the existence of both of such constraints. The following theorem states that even if we have both norm and polyhedral constraints on the disturbances, we still can reduce the final optimization program to one that is can be solved efficiently by standard algorithms such as cutting plane (as in regular structural SVMs).

Theorem 3.6. *For $\Delta^2\Phi(\boldsymbol{x},\boldsymbol{y}) = \{\boldsymbol{\delta} | \|\boldsymbol{M\delta}\| \leq 1, \boldsymbol{A\delta} \leq \boldsymbol{b}\}$, the optimization program of the robust structural SVM in (2) reduces to the following ordinary 1-slack structural SVM:*

$$\begin{aligned} &\underset{\boldsymbol{w},\boldsymbol{\lambda}\geq 0,\zeta}{\text{minimize}}\ Cf(\boldsymbol{w}) + \|\boldsymbol{M}^{-1}(\boldsymbol{w} - \boldsymbol{A}^T\boldsymbol{\lambda})\|^* + \boldsymbol{b}^T\boldsymbol{\lambda} + \zeta \\ &\text{subject to} \quad \zeta \geq \sup_{\tilde{\boldsymbol{y}}} \boldsymbol{w}^T(\boldsymbol{\phi}(\boldsymbol{x},\tilde{\boldsymbol{y}}) - \boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{y})) + \Delta(\boldsymbol{y},\tilde{\boldsymbol{y}}) \end{aligned} \tag{5}$$

The results in Theorems 3.1, 3.5, and 3.6 apply to binary and multi-class SVMs as well simply by restricting the space of y to a small set of values. For 3.1, this reduces to results previous proved for binary SVMs [3]. For the later theorems, we are not aware of any analogous previous work in binary and multi-class SVMs. Some limiting cases of Theorem 3.6, are also interesting. For example, let the polyhedron $\boldsymbol{A\delta} \leq \boldsymbol{b}$ be infinitely large, or equivalently elements of the vector $\boldsymbol{b}$ be infinitely large; then, $\boldsymbol{\lambda}$ must be $\boldsymbol{0}$, which recovers the regularization term $\|\boldsymbol{M}^{-1}\boldsymbol{w}\|^*$ that is introduced in Theorem 3.1. 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 it means that $\boldsymbol{\delta} \to \boldsymbol{0}$ in the robust formulation. Intuitively, it means that no changes can be applied on $(\boldsymbol{\phi}(\boldsymbol{x},\tilde{\boldsymbol{y}}) - \boldsymbol{\phi}(\boldsymbol{x},\boldsymbol{y})$ by the adversary; in that case $\boldsymbol{M}^{-1} \to \boldsymbol{0}$ ($\boldsymbol{M}^{-1}$ 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, resulting in (4).

4 Conclusion

In this paper, we showed that the robust formulation of structural SVMs, which is intractable in general, can be reduced to more tractable optimization programs. Firstly we showed, when the superimposed disturbances in feature space are restricted to an ellipsoid defined by an arbitrary norm, then it is equivalent to regularization of ordinary structural SVMs with the dual norm. Secondly, we showed that it is possible to acquire robustness when the feature variations are restricted by a polyhedron, and finally, we derived a third robust formulation for structural SVMs, where feature variations are restricted by both polyhedral and norm constraints.

Acknowledgments

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 author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ARO, NSF, or the U.S. Government.

References

[1] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations research*, 52(1):35–53, 2004.

[2] Dimitris Bertsimas, Dessislava Pachamanova, and Melvyn Sim. Robust linear optimization under general norms. *Operations Research Letters*, 32(6):510–516, 2004.

[3] Huan Xu, Constantine Caramanis, and Shie Mannor. Robustness and regularization of support vector machines. *The Journal of Machine Learning Research*, 10:1485–1510, 2009.

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

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

[6] Roi Livni and Amir Globerson. A simple geometric interpretation of svm using stochastic adversaries. 2013.

[7] MohamadAli Torkamani and Daniel Lowd. Convex adversarial collective classification. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 642–650, 2013.

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

[9] Aharon Ben-Tal and Arkadi Nemirovski. Robust solutions of uncertain linear programs. *Operations research letters*, 25(1):1–13, 1999.

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

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

[12] Gert RG Lanckriet, Laurent El Ghaoui, Chiranjib Bhattacharyya, and Michael I Jordan. A robust minimax approach to classification. *The Journal of Machine Learning Research*, 3:555–582, 2003.

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

[14] Chiranjib Bhattacharyya, KS Pannagadatta, and Alexander J Smola. A second order cone programming formulation for classifying missing data. *Advances in neural information processing systems*, 17:153–160, 2004.

[15] Pannagadatta K Shivaswamy, Chiranjib Bhattacharyya, and Alexander J Smola. Second order cone programming approaches for handling missing and uncertain data. *The Journal of Machine Learning Research*, 7:1283–1314, 2006.

[16] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the twenty-first international conference on Machine learning*, page 104. ACM, 2004.

[17] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6(2):1453, 2006.

[18] Thorsten Joachims, Thomas Finley, and Chun-Nam John Yu. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59, 2009.

[19] D. Sontag, A. Globerson, and T. Jaakkola. Introduction to dual decomposition for inference. 2010.

[20] B. Taskar, M. F. Wong, P. Abbeel, and D. Koller. Max-margin Markov networks. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[21] Ben Taskar, Dan Klein, Michael Collins, Daphne Koller, and Christopher Manning. Max-margin parsing. In *Proc. EMNLP*, pages 1–8, 2004.