14 Robust Optimization in Machine Learning

Constantine Caramanis caramanis@mail.utexas.edu
The University of Texas at Austin
Austin, Texas

Shie Mannor shie@ee.technion.ac.il
Technion, the Israel Institute of Technology
Haifa, Israel

Huan Xu huan.xu@mail.utexas.edu
The University of Texas at Austin
Austin, Texas

Robust optimization is a paradigm that uses ideas from convexity and duality to immunize solutions of convex problems against bounded uncertainty in the parameters of the problem. Machine learning is fundamentally about making decisions under uncertainty, and optimization has long been a central tool; thus, at a high level there is no surprise that robust optimization should have a role to play. Indeed, the first part of the story told in this chapter is about specializing robust optimization to specific optimization problems in machine learning. Yet, beyond this, there have been several surprising and deep developments in the use of robust optimization and machine learning, connecting consistency, generalization ability, and other properties (such as sparsity and stability) to robust optimization.

In addition to surveying the direct applications of robust optimization to machine learning, important in their own right, this chapter explores some of these deeper connections, and points the way toward opportunities for applications and challenges for further research.

14.1 Introduction

Learning, optimization, and decision making from data must cope with uncertainty introduced both implicitly and explicitly. Uncertainty can be explicitly introduced when the data collection process is noisy, or when some data are corrupted. It may be introduced when the model specification is wrong, assumptions are missing, or factors are overlooked. Uncertainty is also implicitly present in pristine data, insofar as a finite sample empirical distribution, or function thereof, cannot exactly describe the true distribution in most cases. In the optimization community, it has long been known that the effect of even small uncertainty can be devastating in terms of the quality or feasibility of a solution. In machine learning, overfitting has long been recognized as a central challenge, and a plethora of techniques, many of them regularization-based, have been developed to combat this problem. The theoretical justification for many of these techniques lies in controlling notions of complexity, such as metric entropy or VC-dimension.

This chapter considers both uncertainty in optimization, and overfitting, from a unified perspective: robust optimization. In addition to introducing a novel technique for designing algorithms that are immune to noise and do not overfit data, robust optimization also provides a theoretical justification for the success of these algorithms: algorithms have certain properties, such as consistency, good generalization, or sparsity, *because they are robust.*

Robust optimization (e.g., Soyster, 1973; El Ghaoui and Lebret, 1997; Ben-Tal and Nemirovski, 2000; Bertsimas and Sim, 2004; Bertsimas et al., 2010; Ben-Tal et al., 2009, and many others) is designed to deal with parameter uncertainty in convex optimization problems. For example, one can imagine a linear program, $\min : \{\mathbf{c}^\top\mathbf{x} \,|\, A\mathbf{x} \leq \mathbf{b}\}$, where there is uncertainty in the constraint matrix A, the objective function, $\mathbf{c}$, or the right-hand-side vector, $\mathbf{b}$. Robust optimization develops immunity to a deterministic or set-based notion of uncertainty. Thus, in the face of uncertainty in A, instead of solving $\min : \{\mathbf{c}^\top\mathbf{x} \,|\, A\mathbf{x} \leq \mathbf{b}\}$, one solves $\min : \{\mathbf{c}^\top\mathbf{x} \,|\, A\mathbf{x} \leq \mathbf{b},\ \forall A \in \mathcal{U}\}$, for some suitably defined *uncertainty set* $\mathcal{U}$. We give a brief introduction to robust optimization in Section 14.2.

The remainder of this chapter is organized as follows. In Section 14.2 we provide a brief review of robust optimization. In Section 14.3 we discuss direct applications of robust optimization to constructing algorithms that are resistant to data corruption. This is a direct application not only of the methodology of robust optimization, but also of the motivation behind the development of robust optimization. The focus is on developing computationally efficient algorithms, resistant to bounded but otherwise arbitrary (even adversarial) noise. In Sections 14.4–14.6, we show that robust optimization's impact on machine learning extends far outside the originally envisioned scope as developed in the optimization literature. In Section 14.4, we show that many existing machine learning algorithms that are based on regularization, including support vector machines (SVMs), ridge regression, and Lasso, are special cases of robust optimization. Using this reinterpretation, their success can be understood from a unified perspective. We also show how the flexibility of robust optimization paves the way for the design of new regularization-like algorithms. Moreover, we show that robustness can be used directly to prove properties such as regularity and sparsity. In Section 14.5, we show that robustness can be used to prove statistical consistency. Then, in Section 14.6, we extend the results of Section 14.5, showing that an algorithm's generalization ability and its robustness are related in a fundamental way.

In summary, we show that robust optimization has deep connections to machine learning. In particular it yields a unified paradigm that (a) explains the success of many existing algorithms; (b) provides a prescriptive algorithmic approach to creating new algorithms with desired properties; and (c) allows us to prove general properties of an algorithm.

14.2 Background on Robust Optimization

In this section we provide a brief background on robust optimization, and refer the reader to the survey by Bertsimas et al. (2010), the textbook of Ben-Tal et al. (2009), and references to the original papers therein for more details.

Optimization affected by parameter uncertainty has long been a focus of the mathematical programming community. As has been demonstrated in compelling fashion (Ben-Tal and Nemirovski, 2000), solutions to optimization problems can exhibit remarkable sensitivity to perturbations in the problem parameters, thus often rendering a computed solution highly infeasible, suboptimal, or both. This parallels developments in related fields, particularly robust control (refer to Zhou et al., 1996; Dullerud and Paganini, 2000, and the references therein).

Stochastic programming (e.g., Prékopa, 1995; Kall and Wallace, 1994) assumes that the uncertainty has a probabilistic description. In contrast, robust optimization is built on the premise that the parameters vary arbitrarily in some a priori known bounded set, called the *uncertainty set*. Suppose we are optimizing a function $f_0(\mathbf{x})$, subject to the m constraints $f_i(\mathbf{x}, \mathbf{u}_i) \leq 0$, $i = 1, \ldots, m$, where $\mathbf{u}_i$ denotes the parameters of function i. Then, whereas the nominal optimization problem solves $\min\{f_0(\mathbf{x}) : f_i(\mathbf{x}, \mathbf{u}_i) \leq 0, i = 1, \ldots, m\}$, assuming that the $\mathbf{u}_i$ are known, robust optimization solves

$$\begin{aligned} \min_{\mathbf{x}} : \quad & f_0(\mathbf{x}) \\ \text{s.t.} : \quad & f_i(\mathbf{x}, \mathbf{u}_i) \leq 0, \ \forall \mathbf{u}_i \in \mathcal{U}_i, \ i = 1, \ldots, m. \end{aligned} \tag{14.1}$$

14.2.1 Computational Tractability

The tractability of robust optimization, subject to standard and mild Slater-like regularity conditions, amounts to separation for the convex set: $\mathcal{X}(\mathcal{U}) \triangleq \{\mathbf{x} : f_i(\mathbf{x}, \mathbf{u}_i) \leq 0, \ \forall \mathbf{u}_i \in \mathcal{U}_i, \ i = 1, \ldots, m\}$. If there is an efficient algorithm that asserts $\mathbf{x} \in \mathcal{X}(\mathcal{U})$ or otherwise provides a separating hyperplane, then (14.2) can be solved in polynomial time. While the set $\mathcal{X}(\mathcal{U})$ is a convex set as long as each function f_i is convex in $\mathbf{x}$, it is not in general true that there is an efficient separation algorithm for the set $\mathcal{X}(\mathcal{U})$. However, in many cases of broad interest and application, solving the robust problem can be done efficiently—the robustified problem may be of complexity comparable to that of the nominal one. We outline some of the main complexity results below.

14.2.1.1 An Example: Linear Programs with Polyhedral Uncertainty

When the uncertainty set, $\mathcal{U}$, is polyhedral, the separation problem is not only efficiently solvable, it is also in fact linear; thus the robust counterpart is equivalent to a linear optimization problem. To illustrate this, consider the problem with uncertainty in the constraint matrix:

$$\begin{aligned} \min_{\mathbf{x}} : \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} : \quad & \max_{\{\mathbf{a}_i : \mathbf{D}_i \mathbf{a}_i \leq \mathbf{d}_i\}} [\mathbf{a}_i^\top \mathbf{x}] \leq b_i, \quad i = 1, \ldots, m. \end{aligned}$$

The dual of the subproblem (recall that $\mathbf{x}$ is not a variable of optimization in the inner max) again becomes a linear program

$$\begin{bmatrix} \max_{\mathbf{a}_i} : & \mathbf{a}_i^\top \mathbf{x} \\ \text{s.t.} : & \mathbf{D}_i \mathbf{a}_i \leq \mathbf{d}_i \end{bmatrix} \longleftrightarrow \begin{bmatrix} \min_{\mathbf{p}_i} : & \mathbf{p}_i^\top \mathbf{d}_i \\ \text{s.t.} : & \mathbf{p}_i^\top \mathbf{D}_i = \mathbf{x} \\ & \mathbf{p}_i \geq 0 \end{bmatrix},$$

and therefore the robust linear optimization now becomes:

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{p}_1, \ldots, \mathbf{p}_m} : \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} : \quad & \mathbf{p}_i^\top \mathbf{d}_i \leq b_i, \quad i = 1, \ldots, m \\ & \mathbf{p}_i^\top \mathbf{D}_i = \mathbf{x}, \quad i = 1, \ldots, m \\ & \mathbf{p}_i \geq 0, \quad i = 1, \ldots, m. \end{aligned}$$

Thus the size of such problems grows polynomially in the size of the nominal problem and the dimensions of the uncertainty set.

14.2.1.2 Some General Complexity Results

We now list a few of the complexity results that are relevant to this chapter. The reader may refer to Bertsimas et al. (2010); Ben-Tal et al. (2009), and references therein for further details. The robust counterpart for a linear program (LP) with polyhedral uncertainty is again an LP. For an LP with ellipsoidal uncertainty, the counterpart is a second order cone program (SOCP). A convex quadratic program with ellipsoidal uncertainty has a robust counterpart that is a semidefinite program (SDP). An SDP with ellipsoidal uncertainty has an NP-hard robust counterpart.

14.2.2 Probabilistic Interpretations and Results

The computational advantage of robust optimization is largely due to the fact that the formulation is deterministic, and one deals with uncertainty sets rather than probability distributions. While the paradigm makes sense when the disturbances are not stochastic, or the distribution is not known, tractability advantages have made robust optimization an appealing computational framework even when the uncertainty is stochastic and the distribution is fully or partially known. A major success of robust optimization has been the ability to derive a priori probability guarantees—for instance, probability of feasibility—that the solution to a robust optimization will satisfy, under a variety of probabilistic assumptions. Thus robust optimization is a tractable framework one can use to build solutions with probabilistic guarantees such as minimum probability of feasibility, or maximum probability of hinge loss beyond some threshold level, and so on. This probabilistic interpretation of robust optimization is used throughout this chapter.

14.3 Robust Optimization and Adversary Resistant Learning

In this section we overview some of the direct applications of robust optimization to coping with uncertainty (adversarial or stochastic) in machine learning problems. The main themes are (a) the formulations one obtains when using different uncertainty sets and (b) the probabilistic interpretation and results one can derive by using robust optimization. Using ellipsoidal uncertainty, we show that the resulting robust problem is tractable. Moreover, we show that this robust formulation has interesting probabilistic interpretations. Then, using a polyhedral uncertainty set, we show that sometimes it is possible to tractably model combinatorial uncertainty, such as missing data.

Robust optimization-based learning algorithms have been proposed for various learning tasks, such as learning and planning (Nilim and El Ghaoui, 2005), Fisher linear discriminant analysis (Kim et al., 2005), PCA (d'Aspremont et al., 2007), and many others. Instead of providing a comprehensive survey, we use support vector machines (SVMs; e.g., Vapnik and Lerner, 1963; Boser et al., 1992; Cortes and Vapnik, 1995) to illustrate the methodology of robust optimization.

Standard SVMs consider the standard binary classification problem, where we are given a finite number of training samples $\{\mathbf{x}_i, y_i\}_{i=1}^m \subseteq \mathbb{R}^n \times \{-1, +1\}$, and must find a linear classifier, specified by the function $h^{\mathbf{w},b}(\mathbf{x}) = \operatorname{sgn}(\langle \mathbf{w}, \mathbf{x}\rangle + b)$, where $\langle \cdot, \cdot \rangle$ denotes the standard inner product. The parameters $(\mathbf{w}, b)$ are obtained by solving the following convex optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} : \quad & r(\mathbf{w}, b) + C\sum_{i=1}^{m} \xi_i \\ \text{s.t.} : \quad & \xi_i \geq \big[1 - y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b)\big], \quad i = 1, \ldots, m; \\ & \xi_i \geq 0, \quad i = 1, \ldots, m; \end{aligned} \tag{14.2}$$

where $r(\mathbf{w}, b)$ is a regularization term, e.g., $r(\mathbf{w}, b) = \frac{1}{2}\|\mathbf{w}\|_2^2$. There are a number of related formulations, some focusing on controlling VC-dimension, promoting sparsity, or some other property (see Schölkopf and Smola (2001); Steinwart and Christmann (2008), and references therein).

There are three natural ways that uncertainty affects the input data: corruption in the location, $\mathbf{x}_i$; corruption in the label, y_i; and corruption via altogether missing data. We outline some applications of robust optimization to these three settings.

14.3.1 Corrupted Location

Given observed points $\{\mathbf{x}_i\}$, the additive uncertainty model assumes that $\mathbf{x}_i^{true} = \mathbf{x}_i + \mathbf{u}_i$. Robust optimization protects against the uncertainty $\mathbf{u}_i$ by minimizing the regularized training loss on all possible locations of the $\mathbf{u}_i$ in some uncertainty set, $\mathcal{U}_i$.

Trafalis and Gilbert (2007) consider the ellipsoidal uncertainty set given by

$$\mathcal{U}_i = \big\{\mathbf{u}_i : \mathbf{u}_i^\top \Sigma_i \mathbf{u}_i \leq 1\big\}, \quad i = 1, \ldots, m,$$

so that each constraint becomes $\xi_i \geq \left[1 - y_i(\langle \mathbf{w}, \mathbf{x}_i + \mathbf{u}_i \rangle + b)\right]$, $\forall \mathbf{u}_i \in \mathcal{U}_i$. By duality, this is equivalent to $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 + \|\Sigma_i^{1/2}\mathbf{w}\|_2 - \xi_i$, and hence their version of robust SVM reduces to

$$\begin{aligned}
\min_{\mathbf{w},b,\xi} : \quad & r(\mathbf{w}, b) + C\sum_{i=1}^{m} \xi_i \\
\text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i + \|\Sigma_i^{1/2}\mathbf{w}\|_2; \quad i = 1, \ldots, m; \\
& \xi_i \geq 0; \quad i = 1, \ldots, m.
\end{aligned} \tag{14.3}$$

Trafalis and Gilbert (2007) use $r(\mathbf{w}, b) = \frac{1}{2}\|\mathbf{w}\|_2$, while Bhattacharyya et al. (2004) use the sparsity-inducing regularizer $r(\mathbf{w}, b) = \|\mathbf{w}\|_1$. In both settings, the robust problem is an instance of a second-order cone program (SOCP). Available solvers can solve SOCPs with hundreds of thousands of variables and more.

If the uncertainty $\mathbf{u}_i$ is stochastic, one can use this robust formulation to find a classifier that satisfies constraints on the probability (w.r.t. the distribution of $\mathbf{u}_i$) that each constraint is violated. In Shivaswamy et al. (2006), the authors consider two varieties of such chance constraints for $i = 1, \ldots, m$:

$$\begin{aligned}
(a) \quad & \Pr_{\mathbf{u}_i \sim \mathcal{N}(\tilde{\mathbf{0}}, \Sigma_i)}\left(y_i(\mathbf{w}^\top(\mathbf{x}_i + \mathbf{u}_i) + b) \geq 1 - \xi_i\right) \geq 1 - \kappa_i; \\
(b) \quad & \inf_{\mathbf{u}_i \sim (\tilde{\mathbf{0}}, \Sigma_i)} \Pr_{\mathbf{u}_i}\left(y_i(\mathbf{w}^\top(\mathbf{x}_i + \mathbf{u}_i) + b) \geq 1 - \xi_i\right) \geq 1 - \kappa_i.
\end{aligned} \tag{14.4}$$

Constraint (a) controls the probability of constraint violation when the uncertainty follows a known Gaussian distribution. Constraint (b) is more conservative: it controls the worst-case probability of constraint violation, over all centered distributions with variance Σ_i. Theorem 14.1 says that the robust formulation with ellipsoidal uncertainty sets as above can be used to control both of these quantities.

Theorem 14.1. *For $i = 1, \ldots, m$ consider the robust constraint as given above:*

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i + \gamma_i\|\Sigma^{1/2}\mathbf{w}\|_2.$$

If we take $\gamma_i = \Phi^{-1}(\kappa_i)$, for Φ the Gaussian c.d.f., this constraint is equivalent to constraint (a) of (14.4), while taking $\gamma_i = \sqrt{\kappa_i/(1 - \kappa_i)}$ yields constraint (b).

14.3.2 Missing Data

Globerson and Roweis (2006) use robust optimization with polyhedral uncertainty set to address the problem where some of the features of the testing samples may be deleted (possibly in an adversarial fashion). Using a dummy feature to remove the bias term b if necessary, we can rewrite the nominal problem as

$$\min_{\mathbf{w}} : \ \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{m}[1 - y_i\mathbf{w}^\top\mathbf{x}_i]_+.$$

For a given choice of $\mathbf{w}$, the value of the term $[1-y_i\mathbf{w}^\top\mathbf{x}_i]_+$ in the objective, under an adversarial deletion of K features, becomes

$$\begin{aligned}
\max_{\boldsymbol{\alpha}_i} \quad & [1 - y_i\mathbf{w}^\top(\mathbf{x}_i \circ (1 - \boldsymbol{\alpha}_i))]_+ \\
\text{s.t:} \quad & \alpha_{ij} \in \{0,1\}; \quad j = 1,\ldots,n; \\
& \sum_{j=1}^{n}\alpha_{ij} = K,
\end{aligned}$$

where $\circ$ denotes pointwise vector multiplication. While this optimization problem is combinatorial, relaxing the integer constraint $\alpha_{ij} \in \{0,1\}$ to be $0 \le \alpha_{ij} \le 1$ does not change the objective value. Thus, taking the dual of the maximization and substituting into the original problem, one obtains the classifier that is maximally resistant to up to K missing features:

$$\begin{aligned}
\min_{\mathbf{w},\mathbf{v}_i,z_i,t_i,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{m}\xi_i \\
\text{s.t.} \quad & y_i\mathbf{w}^\top\mathbf{x}_i - t_i \ge 1 - \xi_i; \quad i = 1,\ldots,m; \\
& \xi_i \ge 0; \quad i = 1,\ldots,m; \\
& t_i \ge Kz_i + \sum_{j=1}^{n} v_{ij}; \quad i = 1,\ldots,m; \\
& \mathbf{v}_i \ge \mathbf{0}; \quad i = 1,\ldots,m; \\
& z_i + v_{ij} \ge y_i x_{ij} w_{ij}; \quad i = 1,\ldots,m; \quad j = 1,\ldots n.
\end{aligned}$$

This is again an SOCP, and hence fairly large instances can be solved with specialized software.

14.3.3 Corrupted Labels

When the labels are corrupted, the problem becomes more difficult to address due to its combinatorial nature. However, it too has been recently addressed using robust optimization (Caramanis and Mannor, 2008). While there is still a combinatorial price to pay in the complexity of the classifier class, robust optimization can be used to find the optimal classifier; see Caramanis and Mannor (2008) for details.

14.4 Robust Optimization and Regularization

In this section and sections 14.5 and 14.6, we demonstrate that robustness can provide a unified explanation for many desirable properties of a learning algorithm, from regularity and sparsity, to consistency and generalization. A main message of this chapter is that many regularized problems exhibit a "hidden robustness"—they are in fact equivalent to a robust optimization problem—which can then be used to directly prove properties such as consistency and sparsity, and also to design new algorithms. The main problems that highlight this equivalence are regularized support vector machines, ℓ_2-regularized regression, and ℓ_1-regularized regression, also known as Lasso.

14.4.1 Support Vector Machines

We consider regularized SVMs, and show that they are algebraically equivalent to a robust optimization problem. We use this equivalence to provide a probabilistic interpretation of SVMs, which allows us to propose new probabilistic SVM-type formulations. This section is based on Xu et al. (2009).

At a high level it is known that regularization and robust optimization are related; see for instance, El Ghaoui and Lebret (1997), Anthony and Bartlett (1999), and Section 14.3. Yet, the precise connection between robustness and regularized SVMs did not appear until Xu et al. (2009). One of the mottos of robust optimization is to harness the consequences of probability theory without paying the computational cost of having to use its axioms. Consider the additive uncertainty model from Section 14.3.1: $\mathbf{x}_i+\mathbf{u}_i$. If the uncertainties $\mathbf{u}_i$ are stochastic, various limit results (LLN, CLT, etc.) promise that even independent variables will exhibit strong aggregate coupling behavior. For instance, the set $\{(\mathbf{u}_1, \ldots, \mathbf{u}_m) : \sum_{i=1}^m \|\mathbf{u}_i\| \leq c\}$ will have increasing probability as m grows. This motivates designing uncertainty sets with this kind of coupling across uncertainty parameters. We leave it to the reader to check that the *constraint-wise* robustness formulations of Section 14.3.1 cannot be made to capture such coupling constraints across the disturbances $\{\mathbf{u}_i\}$.

We rewrite SVM without slack variables, as an unconstrained optimization. The natural robust formulation now becomes

$$\min_{\mathbf{w},b}\max_{\boldsymbol{u}\in\mathcal{U}}\{r(\mathbf{w},b)+\sum_{i=1}^{m}\max\left[1-y_i(\langle\mathbf{w},\mathbf{x}_i-\boldsymbol{u}_i\rangle+b),0\right]\}, \tag{14.5}$$

where $\boldsymbol{u}$ denotes the collection of uncertainty vectors, $\{\mathbf{u}_i\}$. Describing our coupled uncertainty set requires a few definitions. Definition 14.1 characterizes the effect of different uncertainty sets, and captures the coupling that they exhibit. As an immediate consequence we obtain an equivalent robust optimization formulation for regularized SVMs.

Definition 14.1. *A set* $\mathcal{U}_0 \subseteq \mathbb{R}^n$ *is called an* atomic uncertainty set *if*

(I) $\mathbf{0} \in \mathcal{U}_0$;

(II) *For any* $\mathbf{w}_0 \in \mathbb{R}^n : \sup_{\boldsymbol{u}\in\mathcal{U}_0} [\mathbf{w}_0^\top \boldsymbol{u}] = \sup_{\boldsymbol{u}'\in\mathcal{U}_0} [-\mathbf{w}_0^\top \boldsymbol{u}'] < +\infty.$

Definition 14.2. *Let* $\mathcal{U}_0$ *be an atomic uncertainty set. A set* $\mathcal{U} \subseteq \mathbb{R}^{n\times m}$ *is called a* sublinear aggregated uncertainty set *of* $\mathcal{U}_0$*, if*

$$\mathcal{U}^- \subseteq \mathcal{U} \subseteq \mathcal{U}^+,$$

$$\text{where} \quad \mathcal{U}^- \triangleq \bigcup_{t=1}^{m} \mathcal{U}_t^-; \qquad \mathcal{U}_t^- \triangleq \{(\boldsymbol{u}_1,\dots,\boldsymbol{u}_m)|\boldsymbol{u}_t \in \mathcal{U}_0;\ \boldsymbol{u}_{i\neq t} = \mathbf{0}\}.$$

$$\mathcal{U}^+ \triangleq \{(\alpha_1\boldsymbol{u}_1,\dots,\alpha_m\boldsymbol{u}_m)| \sum_{i=1}^{m} \alpha_i = 1;\ \alpha_i \geq 0,\ \boldsymbol{u}_i \in \mathcal{U}_0,\ i=1,\dots,m\}.$$

Sublinear aggregated uncertainty models the case where the disturbances on each sample are treated identically, but their aggregate behavior across multiple samples is controlled. Some interesting examples include

(1) $\mathcal{U} = \{(\boldsymbol{u}_1,\dots,\boldsymbol{u}_m)| \sum_{i=1}^{m} \|\boldsymbol{u}_i\| \leq c\}$;

(2) $\mathcal{U} = \{(\boldsymbol{u}_1,\dots,\boldsymbol{u}_m)|\exists t \in [1:m];\ \|\boldsymbol{u}_t\| \leq c;\ \boldsymbol{u}_i = \mathbf{0}, \forall i \neq t\}$; and

(3) $\mathcal{U} = \{(\boldsymbol{u}_1,\dots,\boldsymbol{u}_m)| \sum_{i=1}^{m} \sqrt{c\|\boldsymbol{u}_i\|} \leq c\}$.

All these examples share the same atomic uncertainty set $\mathcal{U}_0 = \{\boldsymbol{u} \big| \|\boldsymbol{u}\| \leq c\}$. Figure 14.1 illustrates a sublinear aggregated uncertainty set for $n = 1$ and $m = 2$, that is, the training set consists of two univariate samples.

Theorem 14.2. *Assume* $\{\mathbf{x}_i, y_i\}_{i=1}^m$ *are nonseparable,* $r(\cdot,\cdot) : \mathbb{R}^{n+1} \to \mathbb{R}$ *is an arbitrary function, and* $\mathcal{U}$ *is a sublinear aggregated uncertainty set with corresponding atomic uncertainty set* $\mathcal{U}_0$*. Then the min-max problem*

$$\min_{\mathbf{w},b} \sup_{(\boldsymbol{u}_1,\dots,\boldsymbol{u}_m)\in\mathcal{U}} \left\{ r(\mathbf{w},b) + \sum_{i=1}^{m} \max\left[1 - y_i(\langle \mathbf{w}, \mathbf{x}_i - \boldsymbol{u}_i\rangle + b),\, 0\right] \right\} \tag{14.6}$$

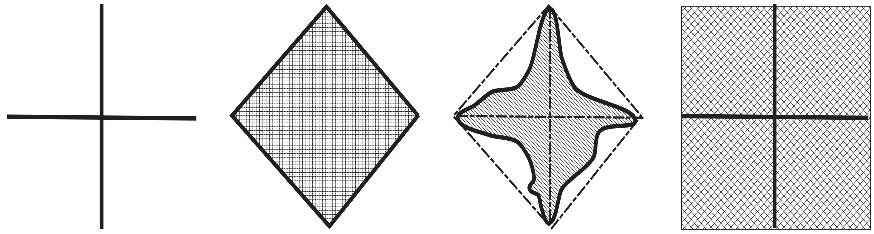

Figure 14.1: A sublinear aggregated uncertainty set $\mathcal{U}$, and its contrast with the box uncertainty set.

is equivalent to the following optimization problem on $\mathbf{w}, b, \boldsymbol{\xi}$*:*

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} : \quad & r(\mathbf{w}, b) + \sup_{\boldsymbol{u} \in \mathcal{U}_0} (\mathbf{w}^\top \boldsymbol{u}) + \sum_{i=1}^{m} \xi_i, \\ \text{s.t.} : \quad & \xi_i \geq 1 - [y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b)], \quad i = 1, \ldots, m; \\ & \xi_i \geq 0, \quad i = 1, \ldots, m. \end{aligned} \tag{14.7}$$

The minimization of (14.7) is attainable when $r(\cdot, \cdot)$ *is lower semi-continuous.*

Proof. We give only the proof idea. The details can be found in Xu et al. (2009). Define

$$v(\mathbf{w}, b) \quad \triangleq \quad \sup_{\boldsymbol{u} \in \mathcal{U}_0} (\mathbf{w}^\top \boldsymbol{u}) + \sum_{i=1}^{m} \max \left[1 - y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b), 0\right].$$

In the first step, we show

$$v(\hat{\mathbf{w}}, \hat{b}) \leq \sup_{(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m) \in \mathcal{U}^-} \sum_{i=1}^{m} \max \left[1 - y_i(\langle \hat{\mathbf{w}}, \mathbf{x}_i - \boldsymbol{u}_i \rangle + \hat{b}), 0\right]. \tag{14.8}$$

This follows because the samples are nonseparable. In the second step, we prove the reverse inequality:

$$\sup_{(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m) \in \mathcal{U}^+} \sum_{i=1}^{m} \max \left[1 - y_i(\langle \hat{\mathbf{w}}, \mathbf{x}_i - \boldsymbol{u}_i \rangle + \hat{b}), 0\right] \leq v(\hat{\mathbf{w}}, \hat{b}). \tag{14.9}$$

This holds regardless of separability. Combining the two, adding the regularizer, and then infimizing both sides concludes the proof. □

An immediate corollary is that a special case of our robust formulation is equivalent to the norm-regularized SVM setup:

Corollary 14.3. *Let* $\mathcal{T} \triangleq \left\{(\boldsymbol{u}_1, \ldots \boldsymbol{u}_m) | \sum_{i=1}^{m} \|\boldsymbol{u}_i\|^* \leq c\right\}$*, where* $\|\cdot\|^*$

stands for the dual norm of $\|\cdot\|$*. If the training samples* $\{\mathbf{x}_i, y_i\}_{i=1}^m$ *are nonseparable, then the following two optimization problems on* $(\mathbf{w}, b)$ *are equivalent.*

$$\min_{\mathbf{w},b}: \quad \max_{(\boldsymbol{u}_1,\ldots,\boldsymbol{u}_m)\in\mathcal{T}} \sum_{i=1}^m \max\left[1 - y_i\big(\langle \mathbf{w},\, \mathbf{x}_i - \boldsymbol{u}_i\rangle + b\big), 0\right], \quad (14.10)$$

$$\min_{\mathbf{w},b}: \quad c\|\mathbf{w}\| + \sum_{i=1}^m \max\left[1 - y_i\big(\langle \mathbf{w},\, \mathbf{x}_i\rangle + b\big), 0\right]. \quad (14.11)$$

Proof. Let $\mathcal{U}_0$ be the dual-norm ball $\{\boldsymbol{u} | \|\boldsymbol{u}\|^* \leq c\}$ and $r(\mathbf{w}, b) \equiv 0$. Then $\sup_{\|\boldsymbol{u}\|^* \leq c}(\mathbf{w}^\top \boldsymbol{u}) = c\|\mathbf{w}\|$. The corollary follows from Theorem 14.2. Notice that the equivalence holds for any $\mathbf{w}$ and b. □

This corollary explains the common belief that regularized classifiers tend to be more robust. Specifically, it explains the observation that when the disturbance is noise like and neutral rather than adversarial, a norm-regularized classifier (without explicit robustness) has a performance often superior to a *box-type* robust classifier (see Trafalis and Gilbert, 2007). One take-away message is that while robust optimization is adversarial in its formulation, it can be quite flexible, and can be designed to yield solutions, such as the regularized solution above, that are appropriate for a non-adversarial setting.

One interesting research direction is to use this equivalence to find good regularizers without the need for cross validation. This could be done by mapping a measure of the variation in the training data to an appropriate uncertainty set, and then using the above equivalence to map back to a regularizer.

14.4.1.1 Kernelization

The previous results can easily be generalized to the kernelized setting. The kernelized SVM formulation considers a linear classifier in the feature space $\mathcal{H}$, a Hilbert space containing the range of some feature mapping $\Phi(\cdot)$. The standard formulation is as follows,

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}}: \quad & r(\mathbf{w}, b) + \sum_{i=1}^m \xi_i \\ \text{s.t.}: \quad & \xi_i \geq \left[1 - y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b)\right], \quad i = 1, \ldots, m; \\ & \xi_i \geq 0, \quad i = 1, \ldots, m; \end{aligned}$$

where we use the representer theorem (see Schölkopf and Smola (2001)).

The definitions of an atomic uncertainty set and a sublinear aggregated uncertainty set in the feature space are identical to Definitions 14.1 and 14.2, with $\mathbb{R}^n$ replaced by $\mathcal{H}$. Theorem 14.4 is a feature space counterpart of Theorem 14.2, and the proof follows from a similar argument.

Theorem 14.4. *Assume* $\{\Phi(\mathbf{x}_i), y_i\}_{i=1}^m$ *are not linearly separable,* $r(\cdot) : \mathcal{H} \times \mathbb{R} \to \mathbb{R}$ *is an arbitrary function,* $\mathcal{U} \subseteq \mathcal{H}^m$ *is a sublinear aggregated uncertainty set with corresponding atomic uncertainty set* $\mathcal{U}_0 \subseteq \mathcal{H}$*. Then the min-max problem*

$$\min_{\mathbf{w},b} \sup_{(\boldsymbol{u}_1,\ldots,\boldsymbol{u}_m)\in\mathcal{U}} \left\{ r(\mathbf{w},b) + \sum_{i=1}^m \max\left[1 - y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i) - \boldsymbol{u}_i\rangle + b), 0\right] \right\}$$

is equivalent to

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi}} : \quad & r(\mathbf{w},b) + \sup_{\boldsymbol{u}\in\mathcal{U}_0} (\langle \mathbf{w}, \boldsymbol{u}\rangle) + \sum_{i=1}^m \xi_i, \\ \text{s.t.} : \quad & \xi_i \geq 1 - y_i\big(\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b\big), \quad i = 1,\ldots,m; \\ & \xi_i \geq 0, \quad i = 1,\ldots,m. \end{aligned} \tag{14.12}$$

The minimization of (14.12) is attainable when $r(\cdot,\cdot)$ *is lower semi-continuous.*

For some widely used feature mappings (e.g., RKHS of a Gaussian kernel), $\{\Phi(\mathbf{x}_i), y_i\}_{i=1}^m$ are always separable. In this case, the equivalence reduces to a bound.

Corollary 14.5 is the feature space counterpart of Corollary 14.3, where $\|\cdot\|_{\mathcal{H}}$ stands for the RKHS norm, that is, for $\mathbf{z} \in \mathcal{H}$, $\|\mathbf{z}\|_{\mathcal{H}} = \sqrt{\langle \mathbf{z}, \mathbf{z}\rangle}$.

Corollary 14.5. *Let* $\mathcal{T}_{\mathcal{H}} \triangleq \left\{(\boldsymbol{u}_1, \ldots \boldsymbol{u}_m) \middle| \sum_{i=1}^m \|\boldsymbol{u}_i\|_{\mathcal{H}} \leq c\right\}$*. If* $\{\Phi(\mathbf{x}_i), y_i\}_{i=1}^m$ *are non-separable, then the following two optimization problems on* $(\mathbf{w}, b)$ *are equivalent:*

$$\begin{aligned} \min_{\mathbf{w},b} : \quad & \max_{(\boldsymbol{u}_1,\ldots,\boldsymbol{u}_m)\in\mathcal{T}_{\mathcal{H}}} \sum_{i=1}^m \max\left[1 - y_i\big(\langle \mathbf{w}, \Phi(\mathbf{x}_i) - \boldsymbol{u}_i\rangle + b\big), 0\right], \\ \min_{\mathbf{w},b} : \quad & c\|\mathbf{w}\|_{\mathcal{H}} + \sum_{i=1}^m \max\left[1 - y_i\big(\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b\big), 0\right]. \end{aligned} \tag{14.13}$$

Equation (14.13) is a variant form of the standard SVM that has a squared RKHS norm regularization term, and by convexity arguments the two formulations are equivalent up to a change of tradeoff parameter c. Therefore, Corollary 14.5 essentially means that the standard kernelized SVM is implicitly a robust classifier (without regularization) with disturbance in the featurespace, and the sum of the magnitudes of the disturbance is bounded.

Disturbance in the feature space is less intuitive than disturbance in the sample space, and Lemma 14.6 relates these two different notions.

Lemma 14.6. *Suppose there exists $\mathcal{X} \subseteq \mathbb{R}^n$, $\rho > 0$, and a continuous non-decreasing function $f : \mathbb{R}^+ \to \mathbb{R}^+$ satisfying $f(0) = 0$, such that*

$$k(\mathbf{x}, \mathbf{x}) + k(\mathbf{x}', \mathbf{x}') - 2k(\mathbf{x}, \mathbf{x}') \leq f(\|\mathbf{x} - \mathbf{x}'\|_2^2), \quad \forall \mathbf{x}, \mathbf{x}' \in \mathcal{X}, \|\mathbf{x} - \mathbf{x}'\|_2 \leq \rho.$$

Then,

$$\|\Phi(\hat{\mathbf{x}} + \boldsymbol{u}) - \Phi(\hat{\mathbf{x}})\|_{\mathcal{H}} \leq \sqrt{f(\|\boldsymbol{u}\|_2^2)}, \quad \forall \|\boldsymbol{u}\|_2 \leq \rho, \ \hat{\mathbf{x}}, \hat{\mathbf{x}} + \boldsymbol{\delta} \in \mathcal{X}.$$

Lemma 14.6 essentially says that under certain conditions, robustness in the feature space is a stronger requirement than robustness in the sample space. Therefore, a classifier that achieves robustness in the feature space also achieves robustness in the sample space. Notice that the condition of Lemma 14.6 is rather weak. In particular, it holds for any continuous $k(\cdot, \cdot)$ and bounded domain $\mathcal{X}$.

14.4.1.2 Probabilistic Interpretations

As discussed and demonstrated above, robust optimization can often be used for *probabilistic analysis*. In this section, we show that robust optimization and the equivalence theorem can be used to construct a classifier with *probabilistic margin protection*, that is, a classifier with probabilistic constraints on the chance of violation beyond a given threshold. Second, we show that in the Bayesian setup, if one has a prior only on the total magnitude of the disturbance vector, robust optimization can be used to tune the regularizer.

Probabilistic Protection. We can use Problem (14.6) to obtain an upper bound for a chance-constrained classifier. Suppose the disturbance is stochastic with known distribution. We denote the disturbance vector by $(\boldsymbol{u}_1^r, \ldots \boldsymbol{u}_m^r)$ to emphasize that it is now a random variable. The chance-constrained classifier minimizes the hinge loss that occurs with probability above some given confidence level $\eta \in [0, 1]$. The classifier is given by the optimization problem

$$\begin{aligned} \min_{\mathbf{w}, b, l} : \quad & l \\ \text{s.t.} : \quad & \mathbb{P}\Big\{ \sum_{i=1}^m \max\big[1 - y_i(\langle \mathbf{w}, \mathbf{x}_i - \boldsymbol{u}_i^r \rangle + b), 0\big] \leq l \Big\} \geq 1 - \eta. \end{aligned} \tag{14.14}$$

The constraint controls the η-quantile of the average (or equivalently the sum of) empirical errors. In Shivaswamy et al. (2006), Lanckriet et al. (2003), and Bhattacharyya et al. (2004), the authors explore a different direction; starting from the constraint formulation of SVM as in (14.2), they impose probabilistic constraints on each random variable individually. This formulation requires all constraints to be satisfied with high probability *simultaneously*. Thus, instead of controlling the η-quantile of the average loss, they control the η-quantile of the hinge loss for each sample. For the same reason that box uncertainty in the robust setting may be too conservative, this constraint-wise formulation may also be too conservative.

Problem (14.14) is generally intractable. However, we can approximate it as follows. Let

$$\hat{c} \triangleq \inf\{\alpha \,|\, \mathbb{P}(\sum_{i=1}^{m} \|\boldsymbol{u}_i\|^* \leq \alpha) \geq 1-\eta\}.$$

Notice that $\hat{c}$ is easily simulated, given μ. Then for any $(\mathbf{w}, b)$, with probability no less than $1-\eta$, the following holds:

$$\begin{aligned} &\sum_{i=1}^{m} \max\big[1 - y_i(\langle \mathbf{w},\, \mathbf{x}_i - \boldsymbol{u}_i^r\rangle + b), 0\big] \\ \leq \quad & \max_{\sum_i \|\boldsymbol{u}_i\|^* \leq \hat{c}} \sum_{i=1}^{m} \max\big[1 - y_i(\langle \mathbf{w},\, \mathbf{x}_i - \boldsymbol{u}_i\rangle + b), 0\big]. \end{aligned}$$

Thus (14.14) is upper-bounded by (14.11) with $c = \hat{c}$. This gives an additional probabilistic robustness property of the standard regularized classifier. We observe that we can follow a similar approach using the constraint-wise robust setup, that is, the box uncertainty set. The interested reader can check that this would lead to considerably more pessimistic approximations of the chance constraint.

A Bayesian Regularizer. Next, we show how the above can be used in a Bayesian setup, to obtain an appropriate regularization coefficient. Suppose the total disturbance $c^r \triangleq \sum_{i=1}^{m} \|\boldsymbol{u}_i^r\|^*$ is a random variable and follows a prior distribution $\rho(\cdot)$. This can model, for example, the case where the training sample set is a mixture of several data sets in which the disturbance magnitude of each set is known. Such a setup leads to the following classifier which minimizes the Bayesian (robust) error:

$$\min_{\mathbf{w},b}: \quad \int \Big\{ \max_{\sum \|\boldsymbol{\delta}_i\|^* \leq c} \sum_{i=1}^{m} \max\big[1 - y_i\big(\langle \mathbf{w},\, \mathbf{x}_i - \boldsymbol{u}_i\rangle + b\big), 0\big]\Big\}\, d\rho(c) \qquad (14.15)$$

By Corollary 14.3, the Bayesian classifier (14.15) is equivalent to

$$\min_{\mathbf{w},b}: \quad \int \Big\{ c\|\mathbf{w}\| + \sum_{i=1}^{m} \max\big[1 - y_i\big(\langle \mathbf{w},\, \mathbf{x}_i\rangle + b\big), 0\big]\Big\}\, d\rho(c),$$

which can be further simplified as

$$\min_{\mathbf{w},b}: \quad \bar{c}\|\mathbf{w}\| + \sum_{i=1}^{m} \max\big[1 - y_i\big(\langle \mathbf{w},\, \mathbf{x}_i\rangle + b\big), 0\big],$$

where $\bar{c} \triangleq \int c\, d\rho(c)$. This provides a justifiable parameter tuning method different from cross validation: simply using the expected value of c^r.

14.4.2 Tikhonov Regularized ℓ_2-Regression

We now move from classification and SVMs to regression, and show that ℓ_2-regularized regression, like SVM, is equivalent to a robust optimization problem. This equivalence is then used to define new regularization-like algorithms, and also to prove properties of the regularized solution.

Given input-output pairs $\mathbf{x}_i, y_i$ which form the rows of X and the elements of vector $\mathbf{y}$, respectively, the goal is to find a predictor $\boldsymbol{\beta}$ that minimizes the squared loss $\|\mathbf{y} - X\boldsymbol{\beta}\|_2^2$. As is well known, this problem is often notoriously ill-conditioned, and may not have a unique solution. The classical and much-explored remedy has been, as in the SVM case, regularization. Regularizing with an ℓ_2-norm, known in statistics as ridge regression (Hoerl, 1962) and in analysis as Tikhonov regularization (Tikhonov and Arsenin, 1977), solves the problem[1]

$$\min_{\boldsymbol{\beta}}: \quad \|\mathbf{y} - X\boldsymbol{\beta}\|_2 + \lambda\|\boldsymbol{\beta}\|_2. \tag{14.16}$$

The main result of this section states that Tikhonov-regularized regression is the solution to a robust optimization, where X is subject to matrix-disturbance U with a bounded Frobenius norm.

Theorem 14.7. *The robust optimization formulation*

$$\min_{\boldsymbol{\beta}}: \max_{U:\|U\|_F \le \lambda} \|\mathbf{y} - (X + U)\boldsymbol{\beta}\|_2$$

is equivalent to Tikhonov-regularized regression (14.16).

Proof. For any perturbation U, we have $\|\mathbf{y} - (X+U)\boldsymbol{\beta}\|_2 = \|\mathbf{y} - X\boldsymbol{\beta} - U\boldsymbol{\beta}\|_2$. By the triangle inequality and because $\|U\|_F \le \lambda$, we thus have $\|\mathbf{y} - (X + U)\boldsymbol{\beta}\|_2 \le \|\mathbf{y} - X\boldsymbol{\beta}\| + \lambda\|\boldsymbol{\beta}\|_2$. On the other hand, for any given $\boldsymbol{\beta}$, we can choose a rank-1 U so that $U\boldsymbol{\beta}$ is aligned with $(\mathbf{y} - X\boldsymbol{\beta})$, and thus equality is attained. □

1. This problem is equivalent to one where we square the norm, up to a change in the regularization coefficient, λ.

This connection was first explored in the seminal work of El Ghaoui and Lebret (1997). There, they further show that the solution to the robust counterpart is almost as easily determined as that to the nominal problem: one need perform only a line search, in the case where the SVD of A is available. Thus, the computational cost of the robust regression is comparable to the original formulation.

As with SVMs, the "hidden robustness" has several consequences. By changing the uncertainty set, robust optimization allows for a rich class of regularization-like algorithms. Motivated by problems from robust control, El Ghaoui and Lebret (1997) then consider perturbations that have structure, leading to structured robust least-squares problems. They then analyze tractability and approximations to these structured least squares.[2] Finally, they use the robustness equivalence to prove regularity properties of the solution. Refer to El Ghaoui and Lebret (1997) for further details about structured robustness, tractability, and regularity.

14.4.3 Lasso

In this section, we consider a similar problem: ℓ_1-regularized regression, also known as Lasso (Tibshirani, 1996). Lasso has been explored extensively for its remarkable sparsity properties (e.g., Tibshirani, 1996; Bickel et al., 2009; Wainwright, 2009), most recently under the banner of compressed sensing (e.g., Chen et al. (1999); Candès et al. (2006), Candès and Tao (2006); Candès and Tao (2007); Candès and Tao (2008), Donoho (2006), for an incomplete list). Following the theme of this section, we show that the solution to Lasso is the solution to a robust optimization problem. As with Tikhonov regularization, robustness provides a connection of the regularizer to a physical property: protection from noise. This allows a principled selection of the regularizer. Moreover, by considering different uncertainty sets, we obtain generalizations of Lasso. Next, we go on to show that robustness can itself be used as an avenue for exploring properties of the solution. In particular, we show that robustness explains why the solution is sparse—that is, *Lasso is sparse because it is robust.* The analysis and the specific results obtained differ from standard sparsity results, providing different geometric intuition. This section is based on results reported in Xu et al. (2010a), where full proofs to all stated results can be found.

Lasso, or ℓ_1-regularized regression, has a form similar to ridge regression,

2. Note that arbitrary uncertainty sets may lead to intractable problems. This is because the inner maximization in the robust formulation is of a convex function, and hence is nonconvex.

differing only in the regularizer: [3]

$$\min : \|\mathbf{y} - X\boldsymbol{\beta}\|_2 + \lambda\|\boldsymbol{\beta}\|_1.$$

For a general uncertainty set $\mathcal{U}$, using the same notation as in Section 14.4.2, the robust regression formulation becomes

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^m} \quad \max_{U\in\mathcal{U}} \|\mathbf{y} - (X+U)\boldsymbol{\beta}\|_2, \tag{14.17}$$

In the previous section, the uncertainty set was $\mathcal{U} = \{U : \|U\|_F \leq \lambda\}$. We consider a different uncertainty set here. Writing

$$U = \begin{bmatrix} | & | & \cdots & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_m \\ | & | & \cdots & | \end{bmatrix}, \qquad \text{where } (\mathbf{u}_1, \ldots, \mathbf{u}_m) \in \mathcal{U},$$

let the uncertainty set $\mathcal{U}$ have the form

$$\mathcal{U} \quad \triangleq \quad \Big\{(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_m)\Big| \|\boldsymbol{u}_i\|_2 \leq c_i, \;\; i = 1, \cdots, m\Big\}. \tag{14.18}$$

This is a *featurewise uncoupled* uncertainty set: the uncertainty in different features need not satisfy any joint constraints. In contrast, the constraint $\|U\|_F \leq 1$ used in Section 14.4.2 is featurewise coupled. We revisit coupled uncertainty sets below.

Theorem 14.8. *The robust regression problem (14.17) with an uncertainty set of the form (14.18) is equivalent to the following ℓ_1-regularized regression problem:*

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^m} \quad \Big\{\|\mathbf{y} - X\boldsymbol{\beta}\|_2 + \sum_{i=1}^m c_i|\beta_i|\Big\}. \tag{14.19}$$

Proof. Fix $\boldsymbol{\beta}^*$. We prove that $\max_{U\in\mathcal{U}} \|\mathbf{y} - (X+U)\boldsymbol{\beta}^*\|_2 = \|\mathbf{y} - X\boldsymbol{\beta}^*\|_2 + \sum_{i=1}^m c_i|\beta_i^*|$.

The inequality

$$\max_{U\in\mathcal{U}} \|\mathbf{y} - (X+U)\boldsymbol{\beta}^*\|_2 \leq \|\mathbf{y} - X\boldsymbol{\beta}^*\|_2 + \sum_{i=1}^m |\beta_i^*|c_i$$

follows from the triangle inequality, as in our proof in Section 14.4.2. The

3. Again we remark that with a change of regularization parameter, this is equivalent to the more common form appearing with a square outside the norm.

other inequality follows, if we take

$$\mathbf{u} \triangleq \begin{cases} \frac{\mathbf{y}-X\boldsymbol{\beta}^*}{\|\mathbf{y}-X\boldsymbol{\beta}^*\|_2} & \text{if } X\boldsymbol{\beta}^* \neq \mathbf{y}, \\ \text{any vector with unit } \ell_2\text{-norm} & \text{otherwise;} \end{cases}$$

and let

$$\boldsymbol{u}_i^* \triangleq \begin{cases} -c_i \text{sgn}(\beta_i^*)\mathbf{u} & \text{if } x_i^* \neq 0; \\ -c_i \mathbf{u} & \text{otherwise.} \end{cases}$$

□

Taking $c_i = c$ and normalizing $\mathbf{x}_i$ for all i, Problem (14.19) recovers the well-known Lasso (Tibshirani, 1996; Efron et al., 2004).

14.4.3.1 General Uncertainty Sets

Using this equivalence, we generalize to Lasso-like regularization algorithms in two ways: (a) to the case of an arbitrary norm and (b) to the case of coupled uncertainty sets.

Theorem 14.9. *For $\|\cdot\|_a$, an arbitrary norm in the Euclidean space, the robust regression problem*

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^m} \left\{ \max_{U\in\mathcal{U}_a} \|\mathbf{y} - (X+U)\boldsymbol{\beta}\|_a \right\},$$

where

$$\mathcal{U}_a \triangleq \left\{ (\boldsymbol{u}_1, \cdots, \boldsymbol{u}_m) \middle| \|\boldsymbol{u}_i\|_a \leq c_i, \;\; i = 1, \cdots, m \right\},$$

is equivalent to the following regularized regression problem

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^m} \left\{ \|\mathbf{y} - X\boldsymbol{\beta}\|_a + \sum_{i=1}^m c_i|\beta_i| \right\}.$$

We next consider featurewise coupled uncertainty sets. They can be used to incorporate additional information about potential noise in the problem, when available, to limit the conservativeness of the worst-case formulation. Consider the uncertainty set

$$\mathcal{U}' \triangleq \left\{ (\boldsymbol{u}_1, \cdots, \boldsymbol{u}_m) \middle| \; f_j(\|\boldsymbol{u}_1\|_a, \cdots, \|\boldsymbol{u}_m\|_a) \leq 0; \; j = 1, \cdots, k \right\},$$

where each $f_j(\cdot)$ is a convex function. The resulting robust formulation is equivalent to a more general regularization type of problem and, moreover, it is tractable.

Theorem 14.10. *Let $\mathcal{U}'$ be as above, and assume that the set*

$$\mathcal{Z} \triangleq \{\mathbf{z} \in \mathbb{R}^m | f_j(\mathbf{z}) \leq 0,\ j = 1, \cdots, k;\ \mathbf{z} \geq \mathbf{0}\}$$

has a nonempty relative interior. Then the robust regression problem

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^m} \left\{ \max_{U\in\mathcal{U}'} \|\mathbf{y} - (X+U)\boldsymbol{\beta}\|_a \right\}$$

is equivalent to the following regularized regression problem:

$$\min_{\boldsymbol{\lambda}\in\mathbb{R}^k_+, \boldsymbol{\kappa}\in\mathbb{R}^m_+, \boldsymbol{\beta}\in\mathbb{R}^m} \Big\{ \|\mathbf{y} - X\boldsymbol{\beta}\|_a + v(\boldsymbol{\lambda}, \boldsymbol{\kappa}, \boldsymbol{\beta}) \Big\}, \tag{14.20}$$

$$\textit{where } v(\boldsymbol{\lambda}, \boldsymbol{\kappa}, \boldsymbol{\beta}) \triangleq \max_{\mathbf{c}\in\mathbf{R}^m} \Big[(\boldsymbol{\kappa} + |\boldsymbol{\beta}|)^\top \mathbf{c} - \sum_{j=1}^k \lambda_j f_j(\mathbf{c}) \Big]$$

and, in particular, is efficiently solvable.

The next two corollaries are a direct application of Theorem 14.10.

Corollary 14.11. *Suppose*

$$\mathcal{U}' = \Big\{ (\boldsymbol{\delta}_1, \cdots, \boldsymbol{\delta}_m) \Big| \big\| \|\boldsymbol{\delta}_1\|_a, \cdots, \|\boldsymbol{\delta}_m\|_a \big\|_s \leq l \Big\},$$

for arbitrary norms $\|\cdot\|_a$ and $\|\cdot\|_s$. Then the robust problem is equivalent to the regularized regression problem

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^m} \Big\{ \|\mathbf{y} - X\boldsymbol{\beta}\|_a + l\|\boldsymbol{\beta}\|_s^* \Big\},$$

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

This corollary interprets *arbitrary* norm-based regularizers from a robust regression perspective. For example, taking both $\|\cdot\|_\alpha$ and $\|\cdot\|_s$ to be the Euclidean norm, $\mathcal{U}'$ is the set of matrices with bounded Frobenius norm, and Corollary 14.11 recovers Theorem 14.7.

The next corollary considers general polytope uncertainty sets, where the columnwise norm vector of the realizable uncertainty belongs to a polytope. To illustrate the flexibility and potential use of such an uncertainty set, take $\|\cdot\|_a$ to be the ℓ_1-norm and the polytope to be the standard simplex, the resulting uncertainty set consists of matrices with bounded $\|\cdot\|_{2,1}$-norm. This is the ℓ_1-norm of the ℓ_2-norm of the columns, and has numerous applications, including outlier removal (Xu et al., 2010c).

Corollary 14.12. *Suppose*

$$\mathcal{U}' = \Big\{ (\boldsymbol{u}_1, \cdots, \boldsymbol{u}_m) \Big| T\mathbf{c} \leq \mathbf{s};\ \textit{where: } c_j = \|\boldsymbol{u}_j\|_a \Big\},$$

for a given matrix T, vector $\mathbf{s}$, and arbitrary norm $\|\cdot\|_a$. Then the robust regression is equivalent to the following regularized regression problem with variables $\boldsymbol{\beta}$ and $\boldsymbol{\lambda}$:

$$\begin{aligned}
\min_{\boldsymbol{\beta},\boldsymbol{\lambda}} : \quad & \|\mathbf{y} - X\boldsymbol{\beta}\|_a + \mathbf{s}^\top \boldsymbol{\lambda} \\
\text{s.t.} \quad & \boldsymbol{\beta} \le T^\top \boldsymbol{\lambda}; \\
& -\boldsymbol{\beta} \le T^\top \boldsymbol{\lambda}; \\
& \boldsymbol{\lambda} \ge \mathbf{0}.
\end{aligned}$$

14.4.3.2 Sparsity

In this section, we investigate the sparsity properties of robust regression, and show in particular that Lasso is sparse *because it is robust*. This new connection between robustness and sparsity suggests that robustifying with respect to a featurewise independent uncertainty set might be a plausible way to achieve sparsity for other problems.

We show that if there is any perturbation in the uncertainty set that makes some feature *irrelevant*, that is, not contributing to the regression error, then the optimal robust solution puts no weight there. Thus, if the features in an index set $I \subset \{1, \ldots, n\}$ can be perturbed so as to be made irrelevant, then the solution will be supported on the complement, I^c.

To state the main theorem of this section, we introduce some notation. Given an index subset $I \subseteq \{1, \ldots, n\}$, and a matrix U, let U^I denote the restriction of U to feature set I, that is, U^I equals U on each feature indexed by $i \in I$, and is zero elsewhere. Similarly, given a featurewise uncoupled uncertainty set $\mathcal{U}$, let $\mathcal{U}^I$ be the restriction of $\mathcal{U}$ to the feature set I, that is, $\mathcal{U}^I \triangleq \{U^I \,|\, U \in \mathcal{U}\}$. Any element $U \in \mathcal{U}$ can be written as $U^I + U^{I^c}$ (here $I^c \triangleq \{1, \ldots, n\} \setminus I$) with $U^I \in \mathcal{U}^I$ and $U^{I^c} \in \mathcal{U}^{I^c}$.

Theorem 14.13. *The robust regression problem*

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^m} \left\{ \max_{\Delta A \in \mathcal{U}} \|\mathbf{y} - (X + U)\boldsymbol{\beta}\|_2 \right\} \tag{14.21}$$

has a solution supported on an index set I if there exists some perturbation $\tilde{U} \in \mathcal{U}^{I^c}$, such that the robust regression problem

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^m} \left\{ \max_{U \in \mathcal{U}^I} \|\mathbf{y} - (X + \tilde{U} + U)\boldsymbol{\beta}\|_2 \right\} \tag{14.22}$$

has a solution supported on the set I.

Theorem 14.13 is a special case of Theorem 14.13' with $c_j = 0$ for all $j \notin I$.
Theorem 14.13': Let $\boldsymbol{\beta}^*$ be an optimal solution of the robust regression

problem

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^m}\left\{\max_{U\in\mathfrak{U}}\|\mathbf{y}-(X+U)\boldsymbol{\beta}\|_2\right\}, \tag{14.23}$$

and let $I \subseteq \{1,\cdots,m\}$ be such that $\beta_j^* = 0\ \forall\, j \notin I$. Let

$$\tilde{\mathfrak{U}} \triangleq \left\{(\boldsymbol{u}_1,\cdots,\boldsymbol{u}_m)\middle|\|\boldsymbol{u}_i\|_2 \le c_i,\ \ i\in I;\ \ \|\boldsymbol{u}_j\|_2 \le c_j + l_j,\ \ j\notin I\right\}.$$

Then, $\boldsymbol{\beta}^*$ is an optimal solution of

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^m}\left\{\max_{U\in\tilde{\mathfrak{U}}}\|\mathbf{y}-(\tilde{X}+U)\boldsymbol{\beta}\|_2\right\} \tag{14.24}$$

for any $\tilde{X}$ that satisfies $\|\tilde{\mathbf{x}}_j - \mathbf{x}_j\| \le l_j$ for $j \notin I$, and $\tilde{\mathbf{x}}_i = \mathbf{x}_i$ for $i \in I$.

In fact, we can replace the ℓ_2-norm loss with any loss function $f(\cdot)$ which satisfies the condition that if $\beta_j = 0$, X and X' differ only in the jth column, and then $f(\mathbf{y}, X, \boldsymbol{\beta}) = f(\mathbf{y}, X', \boldsymbol{\beta})$. This theorem thus suggests a methodology for constructing sparse algorithms by solving a robust optimization with respect to columnwise uncoupled uncertainty sets.

When we consider ℓ_2-loss, we can translate the condition of a feature being "irrelevant" to a geometric condition: orthogonality. We now use the result of Theorem 14.13 to show that robust regression has a sparse solution as long as an incoherence-type property is satisfied. This result is more in line with the traditional sparsity results, but we note that the geometric reasoning is different, now based on robustness. Specifically, we show that a feature receives zero weight if it is "nearly" (i.e., within an allowable perturbation) orthogonal to the signal and all relevant features.

Theorem 14.14. *Let $c_i = c$ for all i and consider ℓ_2-loss. Suppose that there exists $I \subset \{1,\cdots,m\}$ such that for all $\mathbf{v} \in \mathrm{span}\big(\{\mathbf{x}_i, i \in I\} \bigcup \{\mathbf{y}\}\big)$, $\|\mathbf{v}\| = 1$, we have $\mathbf{v}^\top \mathbf{x}_j \le c$, $\forall j \notin I$. Then there exists an optimal solution $\boldsymbol{\beta}^*$ that satisfies $\beta_j^* = 0$, $\forall j \notin I$.*

The proof proceeds as Theorem 14.13' would suggest: the columns in I^c can be perturbed so that they are made irrelevant, and thus the optimal solution will not be supported there; see Xu et al. (2010a) for details..

14.5 Robustness and Consistency

In this section we explore a fundamental connection between learning and robustness by using robustness properties to re-prove the consistency of kernelized SVM, and then of Lasso. The key difference from the proofs here and those seen elsewhere (e.g., Steinwart, 2005; Steinwart and Christmann, 2008;

Wainwright, 2009) is that we replace the metric entropy, VC-dimension, and stability conditions typically used, with a robustness condition. Thus we conclude that *SVM and Lasso are consistent because they are robust.*

14.5.1 Consistency of SVM

Let $\mathcal{X} \subseteq \mathbb{R}^n$ be bounded, and suppose the training samples $(\mathbf{x}_i, y_i)_{i=1}^{\infty}$ are generated according to an unknown i.i.d. distribution $\mathbb{P}$ supported on $\mathcal{X} \times \{-1, +1\}$. Theorem 14.15 shows that our robust classifier, and thus regularized SVM, asymptotically minimizes an upper bound of the expected classification error and hinge loss as the number of samples increases.

Theorem 14.15. *Let* $K \triangleq \max_{x \in \mathcal{X}} \|x\|_2$. *Then there exists a random sequence* $\{\gamma_{m,c}\}$ *such that*

1. *The following bounds on the Bayes loss and the hinge loss hold uniformly for all* $(\mathbf{w}, b)$*:*

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}}(\mathbf{1}_{y \neq sgn(\langle \mathbf{w},\, \mathbf{x}\rangle + b)}) \leq \gamma_{m,c} + c\|\mathbf{w}\|_2 + \frac{1}{m}\sum_{i=1}^{m} \max\big[1 - y_i(\langle \mathbf{w},\, \mathbf{x}_i\rangle + b), 0\big];$$

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}}\big(\max(1 - y(\langle \mathbf{w},\, \mathbf{x}\rangle + b),\, 0)\big) \leq$$
$$\gamma_{m,c}(1 + K\|\mathbf{w}\|_2 + |b|) + c\|\mathbf{w}\|_2 + \frac{1}{m}\sum_{i=1}^{m} \max\big[1 - y_i(\langle \mathbf{w},\, \mathbf{x}_i\rangle + b), 0\big].$$

2. *For every* $c > 0$, $\lim_{m\to\infty} \gamma_{m,c} = 0$ *almost surely, and the convergence is uniform in* $\mathbb{P}$.

Proof. We outline the basic idea of the proof here; refer to Xu et al. (2009) for the technical details. We consider the testing sample set as a perturbed copy of the training sample set, and measure the magnitude of the perturbation. For testing samples that have "small" perturbations, Corollary 14.3 guarantees that the quantity $c\|\mathbf{w}\|_2 + \frac{1}{m}\sum_{i=1}^{m} \max\big[1 - y_i(\langle \mathbf{w},\, \mathbf{x}_i\rangle + b), 0\big]$ upper-bounds their total loss. Therefore, we only need to show that the fraction of testing samples having "large" perturbations diminishes to prove the theorem. We show this using a balls and bins argument. Partitioning $\mathcal{X} \times \{-1, +1\}$, we match testing and training samples that fall in the same partition. We then use the Bretagnolle-Huber-Carol inequality for multinomial distributions to conclude that the fraction of unmatched points diminishes to zero. □

Based on Theorem 14.15, it can be further shown that the expected classification error of the solutions of SVM converges to the Bayes risk, that is, SVM is consistent.

14.5.2 Consistency of Lasso

In this section, we re-prove the asymptotic consistency of Lasso by using robustness. The basic idea of the consistency proof is as follows. We show that the robust optimization formulation can be seen to have the maximum *expected error* w.r.t. a class of probability measures. This class includes a kernel density estimator, and using this, we show that Lasso is consistent.

14.5.2.1 Robust Optimization and Kernel Density Estimation

En route to proving the consistency of Lasso based on robust optimization, we discuss another result of independent interest. We link robust optimization to worst-case expected utility, that is, the worst-case expectation over a set of measures. For the proofs, and more along this direction, see Xu et al. (2010a,b). Throughout this section, we use $\mathcal{P}$ to represent the set of all probability measures (on Borel σ-algebra) of $\mathbb{R}^{m+1}$.

We first establish a general result on the equivalence between a robust optimization formulation and a worst-case expected utility:

Proposition 14.16. *Given a function $f : \mathbb{R}^{m+1} \to \mathbb{R}$ and Borel sets $\mathcal{Z}_1, \cdots, \mathcal{Z}_n \subseteq \mathbb{R}^{m+1}$, let*

$$\mathcal{P}_n \triangleq \{\mu \in \mathcal{P} | \forall S \subseteq \{1, \cdots, n\} : \mu(\bigcup_{i \in S} \mathcal{Z}_i) \geq |S|/n\}.$$

The following holds:

$$\frac{1}{n} \sum_{i=1}^{n} \sup_{(\mathbf{x}_i, y_i) \in \mathcal{Z}_i} f(\mathbf{x}_i, y_i) = \sup_{\mu \in \mathcal{P}_n} \int_{\mathbb{R}^{m+1}} f(\mathbf{x}, y) d\mu(\mathbf{x}, y).$$

This leads to the Corollary 14.17 for Lasso, which states that for a given solution $\boldsymbol{\beta}$, the robust regression loss over the training data is equal to the worst-case expected *generalization error*.

Corollary 14.17. *Given $\mathbf{y} \in \mathbb{R}^n$, $X \in \mathbb{R}^{n \times m}$, the following equation holds for any $\boldsymbol{\beta} \in \mathbb{R}^m$,*

$$\|\mathbf{y} - X\boldsymbol{\beta}\|_2 + \sqrt{n} c_n (\|\boldsymbol{\beta}\|_1 + 1) = \sup_{\mu \in \hat{\mathcal{P}}(n)} \sqrt{n \int_{\mathbb{R}^{m+1}} (y' - \mathbf{x}'^{\top} \boldsymbol{\beta})^2 d\mu(\mathbf{x}', y')}. \quad (14.25)$$

Where we let x_{ij} and u_{ij} be the (i,j)-entries of X and U, respectively, and

$$\begin{aligned}\hat{\mathcal{P}}(n) &\triangleq \bigcup_{\|\boldsymbol{\sigma}\|_2\le\sqrt{n}c_n;\,\forall i:\|\boldsymbol{u}_i\|_2\le\sqrt{n}c_n} \mathcal{P}_n(X,U,\mathbf{y},\boldsymbol{\sigma});\\ \mathcal{P}_n(X,U,\mathbf{y},\boldsymbol{\sigma}) &\triangleq \{\mu\in\mathcal{P}|\mathcal{Z}_i=[y_i-\sigma_i,y_i+\sigma_i]\times\prod_{j=1}^{m}[x_{ij}-u_{ij},x_{ij}+u_{ij}];\\ &\quad \forall S\subseteq\{1,\cdots,n\}:\mu(\bigcup_{i\in S}\mathcal{Z}_i)\ge|S|/n\}.\end{aligned}$$

The proof of consistency relies on showing that the set $\hat{\mathcal{P}}(n)$ of distributions contains a kernel density estimator. Recall the basic definition: the *kernel density estimator* for a density h in $\mathbb{R}^d$, originally proposed in Rosenblatt (1956) and Parzen (1962), is defined by

$$h_n(\mathbf{x})=(nc_n^d)^{-1}\sum_{i=1}^{n}K\left(\frac{\mathbf{x}-\hat{\mathbf{x}}_i}{c_n}\right),$$

where $\{c_n\}$ is a sequence of positive numbers, $\hat{\mathbf{x}}_i$ are i.i.d. samples generated according to h, and K is a Borel measurable function (kernel) satisfying $K\ge 0$, $\int K=1$. See Devroye and Györfi (1985), Scott (1992), and references therein for detailed discussions. A celebrated property of a kernel density estimator is that it converges in $\mathcal{L}^1$ to h when $c_n\downarrow 0$ and $nc_n^d\uparrow\infty$ (Devroye and Györfi, 1985).

14.5.2.2 Density Estimation and Consistency of Lasso

We now use robustness of Lasso to prove its consistency. Throughout, we use c_n to represent the robustness level c where there are n samples. We take c_n to zero as n grows.

Recall the standard generative model in statistical learning: let $\mathbb{P}$ be a probability measure with bounded support that generates i.i.d. samples $(y_i,\mathbf{x}_i)$, and has a density $f^*(\cdot)$. Denote the set of the first n samples by $\mathcal{S}_n$. Define

$$\begin{aligned}\boldsymbol{\beta}(c_n,\mathcal{S}_n) &\triangleq \arg\min_{\boldsymbol{\beta}}\Big\{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-\mathbf{x}_i^\top\boldsymbol{\beta})^2}+c_n\|\beta\|_1\Big\}\\ &= \arg\min_{\boldsymbol{\beta}}\Big\{\frac{\sqrt{n}}{n}\sqrt{\sum_{i=1}^{n}(y_i-\mathbf{x}_i^\top\boldsymbol{\beta})^2}+c_n\|\beta\|_1\Big\};\\ \boldsymbol{\beta}(\mathbb{P}) &\triangleq \arg\min_{\boldsymbol{\beta}}\Big\{\sqrt{\int_{y,\mathbf{x}}(y-\mathbf{x}^\top\boldsymbol{\beta})^2d\mathbb{P}(y,\mathbf{x})}\Big\}.\end{aligned}$$

In words, $\boldsymbol{\beta}(c_n, \mathcal{S}_n)$ is the solution to Lasso with the tradeoff parameter set to $c_n\sqrt{n}$, and $\boldsymbol{\beta}(\mathbb{P})$ is the "true" optimal solution. We establish that $\boldsymbol{\beta}(c_n, \mathcal{S}_n) \to \boldsymbol{\beta}(\mathbb{P})$ using robustness.

Theorem 14.18. *Let $\{c_n\}$ be such that $c_n \downarrow 0$ and $\lim_{n\to\infty} n(c_n)^{m+1} = \infty$. Suppose there exists a constant H such that $\|\boldsymbol{\beta}(c_n, \mathcal{S}_n)\|_2 \leq H$ for all n. Then,*

$$\lim_{n\to\infty} \sqrt{\int_{y,\mathbf{x}} (y - \mathbf{x}^\top \boldsymbol{\beta}(c_n, \mathcal{S}_n))^2 d\mathbb{P}(y, \mathbf{x})} = \sqrt{\int_{y,\mathbf{x}} (y - \mathbf{x}^\top \boldsymbol{\beta}(\mathbb{P}))^2 d\mathbb{P}(y, \mathbf{x})},$$

almost surely.

We give an outline of the proof; refer to Xu et al. (2010a) for the details. In Section 14.4.3 we showed that Lasso is a special case of robust optimization. Then, in Section 14.5.2.1, we proved that robust optimization is equivalent to a worst-case expectation. The proof follows by showing that the sets $\mathcal{P}_n$, in the worst-case expectation equivalent to Lasso, contain a kernel density estimator. Since these sets shrink, consistency follows.

The assumption that $\|\mathbf{x}(c_n, \mathcal{S}_n)\|_2 \leq H$ can be removed. As in Theorem 14.18, the proof technique rather than the result itself is of interest. We refer the interested reader to Xu et al. (2010a).

14.6 Robustness and Generalization

We have already seen that regularized regression and regularized SVMs are special cases of robust optimization, and hence exhibit robustness to perturbed data. This robustness was used above to show that ridge regression has a Lipschitz solution, that Lasso is sparse, and that SVM and Lasso are consistent. In this section, we show that robustness can be used to control the estimation of the risk (i.e., generalization error) of learning algorithms. The results we describe are based on Xu and Mannor (2010b).

Several approaches have been proposed to bound the deviation of the risk from its empirical measurement, and among these methods, those based on uniform convergence and stability are the most widely used (e.g., Vapnik and Chervonenkis, 1991; Evgeniou et al., 2000; Alon et al., 1997; Bartlett, 1998; Bartlett and Mendelson, 2002; Bartlett et al., 2005; Bousquet and Elisseeff, 2002; Poggio et al., 2004; Mukherjee et al., 2006, and many others). We provide a new, robustness-driven approach to proving generalization bounds.

Whereas in previous sections "robustness" was defined directly in terms of robust optimization, here we abstract this definition. Because we consider abstract algorithms in this section, we introduce some necessary notations, that differ from those in previous sections. We use $\mathcal{Z}$ to denote the set from which each sample is drawn, and $\mathcal{H}$ to denote the hypothesis set. Throughout this section we use $\mathbf{s} \in \mathcal{Z}^m$ to denote the training sample set consisting of m training samples $(s_1, \cdots, s_m)$. A learning algorithm $\mathcal{A}$ is thus a mapping from $\mathcal{Z}^m$ to $\mathcal{H}$. We use $\mathcal{A}_\mathbf{s}$ to represent the hypothesis learned, given training set $\mathbf{s}$. For each hypothesis $h \in \mathcal{H}$ and each point $z \in \mathcal{Z}$, there is an associated loss $l(h, z)$, which is nonnegative and upper-bounded uniformly by a scalar M. In the special case of supervised learning, the sample space can be decomposed as $\mathcal{Z} = \mathcal{Y} \times \mathcal{X}$, and the goal is to learn a mapping from $\mathcal{X}$ to $\mathcal{Y}$, that is, to predict the y-component given the x-component. Hence we use $\mathcal{A}_\mathbf{s}(x)$ to represent the predicted y-component (label) of $x \in \mathcal{X}$ when $\mathcal{A}$ is trained on $\mathbf{s}$. We call $\mathcal{X}$ the input space and $\mathcal{Y}$ the output space. We use $_{|x}$ and $_{|y}$ to denote the x-component and y-component of a point. For example, $s_{i|x}$ is the x-component of s_i. Finally, we use $\mathcal{N}(\varepsilon, T, \rho)$ to denote the ε-covering number of a space T equipped with a metric ρ (see van der Vaart and Wellner, 2000, for a precise definition).

Definition 14.3 says that an algorithm is robust, if we can partition the sample set into finite subsets, such that if a new sample falls into the same subset as a training sample, then the loss of the former is close to the loss of the latter.

Definition 14.3. *Algorithm $\mathcal{A}$ is $(K, \varepsilon(\mathbf{s}))$ robust if $\mathcal{Z}$ can be partitioned into K disjoint sets, denoted by $\{C_i\}_{i=1}^K$, such that $\forall s \in \mathbf{s}$,*

$$s, z \in C_i, \quad \Longrightarrow \quad |l(\mathcal{A}_\mathbf{s}, s) - l(\mathcal{A}_\mathbf{s}, z)| \leq \varepsilon(\mathbf{s}). \tag{14.26}$$

14.6.1 Generalization Properties of Robust Algorithms

In this section we use Definition 14.3 to derive PAC bounds for robust algorithms. Let the sample set $\mathbf{s}$ consist of m i.i.d. samples generated by an unknown distribution μ. Let $\hat{l}(\cdot)$ and $l_{\text{emp}}(\cdot)$ denote the expected error and the training error, respectively. That is,

$$\hat{l}(\mathcal{A}_\mathbf{s}) \triangleq \mathbb{E}_{z\sim\mu} l(\mathcal{A}_\mathbf{s}, z); \quad l_{\text{emp}}(\mathcal{A}_\mathbf{s}) \triangleq \frac{1}{m} \sum_{s_i \in \mathbf{s}} l(\mathcal{A}_\mathbf{s}, s_i).$$

Theorem 14.19. *If $\mathbf{s}$ consists of m i.i.d. samples, the loss function $l(\cdot, \cdot)$ is upper-bounded by M, and $\mathcal{A}$ is $(K, \varepsilon(\mathbf{s}))$-robust, then for any $\delta > 0$, with probability at least $1 - \delta$,*

$$\left|\hat{l}(\mathcal{A}_\mathbf{s}) - l_{\text{emp}}(\mathcal{A}_\mathbf{s})\right| \leq \varepsilon(s) + M\sqrt{\frac{2K \ln 2 + 2\ln(1/\delta)}{m}}.$$

Proof. The proof follows by partitioning the set and using inequalities for multinomial random variables, à la the Bretagnolle-Huber-Carol inequality. □

Theorem 14.19 requires that we fix a K a priori. However, it is often worthwhile to consider adaptive K. For example, in the large-margin classification case, typically the margin is known only after $\mathbf{s}$ is realized. That is, the value of K depends on $\mathbf{s}$. Because of this dependency, we need a generalization bound that holds uniformly for all K.

Corollary 14.20. *If* $\mathbf{s}$ *consists of* m *i.i.d. samples, and* $\mathcal{A}$ *is* $(K, \varepsilon_K(\mathbf{s}))$*-robust for all* $K \geq 1$*, then for any* $\delta > 0$*, with probability at least* $1 - \delta$*,*

$$\left|\hat{l}(\mathcal{A}_{\mathbf{s}}) - l_{\mathrm{emp}}(\mathcal{A}_{\mathbf{s}})\right| \leq \inf_{K \geq 1} \left[\varepsilon_K(s) + M\sqrt{\frac{2K \ln 2 + 2 \ln \frac{K(K+1)}{\delta}}{m}}\right].$$

If $\varepsilon(s)$ does not depend on $\mathbf{s}$, we can sharpen the bound given in Corollary 14.20.

Corollary 14.21. *If* $\mathbf{s}$ *consists of* m *i.i.d. samples, and* $\mathcal{A}$ *is* (K, ε_K)*-robust for all* $K \geq 1$*, then for any* $\delta > 0$*, with probability at least* $1 - \delta$*,*

$$\left|\hat{l}(\mathcal{A}_{\mathbf{s}}) - l_{\mathrm{emp}}(\mathcal{A}_{\mathbf{s}})\right| \leq \inf_{K \geq 1} \left[\varepsilon_K + M\sqrt{\frac{2K \ln 2 + 2 \ln \frac{1}{\delta}}{m}}\right].$$

14.6.2 Examples of Robust Algorithms

In this section we provide some examples of robust algorithms. For the proofs of these examples, refer to Xu and Mannor (2010b,a). Our first example is majority voting (MV) classification (e.g., Devroye et al., 1996, Section 6.3), which partitions the input space $\mathcal{X}$ and labels each partition set according to a majority vote of the training samples belonging to it.

Example 14.1 (majority voting)**.** *Let* $\mathcal{Y} = \{-1, +1\}$*. Partition* $\mathcal{X}$ *to* $\mathcal{C}_1, \cdots, \mathcal{C}_K$*, and use* $\mathcal{C}(x)$ *to denote the set to which* x *belongs. A new sample* $x_a \in \mathcal{X}$ *is labeled*

$$\mathcal{A}_{\mathbf{s}}(x_a) \triangleq \begin{cases} 1, & \text{if } \sum_{s_i \in \mathcal{C}(x_a)} \mathbf{1}(s_{i|y} = 1) \geq \sum_{s_i \in \mathcal{C}(x_a)} \mathbf{1}(s_{i|y} = -1); \\ -1, & \text{otherwise.} \end{cases}$$

If the loss function is the prediction error $l(\mathcal{A}_s, z) = 1_{z_{|y} \neq \mathcal{A}_{\mathbf{s}}(z_{|x})}$*, then MV is* $(2K, 0)$*-robust.*

The MV algorithm has a natural partition of the sample space that makes it robust. Another class of robust algorithms is those that have approximately the same testing loss for testing samples that are close (in the sense of geometric distance) to each other, since we can partition the sample space with norm balls, as in the standard definition of covering numbers (van der Vaart and Wellner, 2000). Theorem 14.22 states that an algorithm is robust if two samples being close implies that they have a similar testing error. Thus, in particular, this means that robustness is weaker than uniform stability (Bousquet and Elisseeff, 2002).

Theorem 14.22. *Fix $\gamma > 0$ and metric ρ of $\mathcal{Z}$. Suppose $\mathcal{A}$ satisfies*

$$|l(\mathcal{A}_{\mathbf{s}}, z_1) - l(\mathcal{A}_{\mathbf{s}}, z_2)| \leq \varepsilon(\mathbf{s}), \quad \forall z_1, z_2 : z_1 \in \mathbf{s},\ \rho(z_1, z_2) \leq \gamma,$$

and $\mathcal{N}(\gamma/2, \mathcal{Z}, \rho) < \infty$. Then $\mathcal{A}$ is $\big(\mathcal{N}(\gamma/2, \mathcal{Z}, \rho),\ \varepsilon(\mathbf{s})\big)$-robust.

Theorem 14.22 leads Example 14.3: if the testing error, given the output of an algorithm, is Lipschitz continuous, then the algorithm is robust.

Example 14.2 (Lipschitz continuous functions)**.** *If $\mathcal{Z}$ is compact w.r.t. metric ρ, and $l(\mathcal{A}_{\mathbf{s}}, \cdot)$ is Lipschitz continuous with Lipschitz constant $c(\mathbf{s})$, that is,*

$$|l(\mathcal{A}_{\mathbf{s}}, z_1) - l(\mathcal{A}_{\mathbf{s}}, z_2)| \leq c(\mathbf{s})\rho(z_1, z_2), \quad \forall z_1, z_2 \in \mathcal{Z},$$

then $\mathcal{A}$ is $\big(\mathcal{N}(\gamma/2, \mathcal{Z}, \rho),\ c(\mathbf{s})\gamma\big)$-robust for all $\gamma > 0$.

Theorem 14.22 also implies that SVM, Lasso, feed-forward neural networks, and PCA are robust, as stated in Examples 14.3–14.6.

Example 14.3 (support vector machines)**.** *Let $\mathcal{X}$ be compact. Consider the standard SVM formulation (Cortes and Vapnik, 1995; Schölkopf and Smola, 2001), as discussed in Sections 14.3 and 14.4.*

$$\begin{aligned} \min_{\mathbf{w}, d} \quad & c\|\mathbf{w}\|_{\mathcal{H}}^2 + \sum_{i=1}^{m} \xi_i \\ \text{s.t.} \quad & 1 - s_{i|y}[\langle \mathbf{w},\ \phi(s_{i|x}) \rangle + d] \leq \xi_i, \quad i = 1, \cdots, m; \\ & \xi_i \geq 0, \quad i = 1, \cdots, m. \end{aligned}$$

Here $\phi(\cdot)$ is a feature mapping, $\|\cdot\|_{\mathcal{H}}$ is its RKHS kernel, and $k(\cdot,\cdot)$ is the kernel function. Let $l(\cdot,\cdot)$ be the hinge loss, that is, $l\big((w,d), z\big) = [1 - z_{|y}(\langle w, \phi(z_{|x}) \rangle + d)]^+$, and define $f_{\mathcal{H}}(\gamma) \triangleq \max_{\mathbf{a},\mathbf{b} \in \mathcal{X}, \|\mathbf{a}-\mathbf{b}\|_2 \leq \gamma} \big(k(\mathbf{a},\mathbf{a}) + k(\mathbf{b},\mathbf{b}) - 2k(\mathbf{a},\mathbf{b})\big)$. If $k(\cdot,\cdot)$ is continuous, then for any $\gamma > 0$, $f_{\mathcal{H}}(\gamma)$ is finite, and SVM is $(2\mathcal{N}(\gamma/2, \mathcal{X}, \|\cdot\|_2), \sqrt{f_{\mathcal{H}}(\gamma)/c})$ robust.

Example 14.4 (Lasso)**.** *Let $\mathcal{Z}$ be compact and the loss function be $l(A_{\mathbf{s}}, z) = |z_{|y} - A_{\mathbf{s}}(z_{|x})|$. Lasso (Tibshirani, 1996), which is the following regression* *formulation:*

$$\min_{\mathbf{w}}: \qquad \frac{1}{m}\sum_{i=1}^{m}(s_{i|y} - \mathbf{w}^\top s_{i|x})^2 + c\|\mathbf{w}\|_1,$$

is $\big(\mathcal{N}(\gamma/2, \mathcal{Z}, \|\cdot\|_\infty), (Y(\mathbf{s})/c+1)\gamma\big)$*-robust for all* $\gamma > 0$*, where* $Y(\mathbf{s}) \triangleq \frac{1}{n}\sum_{i=1}^{n} {s_{i|y}}^2$ *.*

Example 14.5 (Feed-forward neural networks)**.** *Let* $\mathcal{Z}$ *be compact and the loss function be* $l(\mathcal{A}_{\mathbf{s}}, z) = |z_{|y} - \mathcal{A}_{\mathbf{s}}(z_{|x})|$*. Consider the* d*-layer neural network (trained on* $\mathbf{s}$*), which is the following predicting rule, given an input* $x \in \mathcal{X}$

$$\begin{aligned} x^0 &:= z_{|x} \\ \forall v = 1, \cdots, d-1: \qquad x_i^v &:= \sigma\Big(\sum_{j=1}^{N_{v-1}} w_{ij}^{v-1} x_j^{v-1}\Big); \quad i = 1, \cdots, N_v; \\ \mathcal{A}_{\mathbf{s}}(x) &:= \sigma\Big(\sum_{j=1}^{N_{d-1}} w_j^{d-1} x_j^{d-1}\Big); \end{aligned}$$

If there exist α *and* β *such that the* d*-layer neural network satisfying that* $|\sigma(a) - \sigma(b)| \leq \beta|a-b|$*, and* $\sum_{j=1}^{N_v} |w_{ij}^v| \leq \alpha$ *for all* v, i*, then it is* $\big(\mathcal{N}(\gamma/2, \mathcal{Z}, \|\cdot\|_\infty), \alpha^d\beta^d\gamma\big)$*-robust, for all* $\gamma > 0$*.*

In Example 14.5, the number of hidden units in each layer has no effect on the robustness of the algorithm and, consequently, on the bound on the testing error. This indeed agrees with Bartlett (1998), where the author showed (using a different approach based on fat-shattering dimension) that for neural networks, the weight plays a more important role than the number of hidden units.

Example 14.6 considers an unsupervised learning algorithm, namely, the principal component analysis algorithm. We show that it is robust if the sample space is *bounded*. This does not contradict the well-known fact that the principal component analysis is sensitive to outliers which are far from the origin.

Example 14.6 (Principal component analysis (PCA))**.** *Let* $\mathcal{Z} \subset \mathbb{R}^m$ *be such that* $\max_{z\in\mathcal{Z}} \|z\|_2 \leq B$*. If the loss function is* $l((w_1, \cdots, w_d), z) = \sum_{k=1}^{d} (w_k^\top z)^2$*, then finding the first* d *principal components, which solves the* *optimization problem over* d *vectors* $w_1, \cdots, w_d \in \mathbb{R}^m$,

$$\begin{aligned} \max_{\mathbf{w}_1,\cdots,\mathbf{w}_k} \quad & \sum_{i=1}^{m}\sum_{k=1}^{d}(\mathbf{w}_k^\top s_i)^2 \\ \text{s.t.} \quad & \|\mathbf{w}_k\|_2 = 1, \quad k = 1, \cdots, d; \\ & \mathbf{w}_i^\top \mathbf{w}_j = 0, \quad i \neq j. \end{aligned}$$

is $(\mathcal{N}(\gamma/2, \mathcal{Z}, \|\cdot\|_2), 2d\gamma B)$*-robust.*

14.7 Conclusion

The purpose of this chapter has been to hint at the wealth of applications and uses of robust optimization in machine learning. Broadly speaking, there are two main methodological frameworks developed here: robust optimization used as a way to make an optimization-based machine learning algorithm robust to noise; and robust optimization as a fundamental tool for analyzing properties of machine learning algorithms and for constructing algorithms with special properties. The properties we have discussed here include sparsity, consistency and generalization. There are many directions of interest that future work can pursue. We highlight two that we consider of particular interest and promise. The first is learning in the high-dimensional setting, where the dimensionality of the models (or parameter space) is of the same order of magnitude as the number of training samples available. Hidden structure such as sparsity or low rank has offered ways around the challenges of this regime. Robustness and robust optimization may offer clues as to how to develop new tools and new algorithms for this setting. A second direction of interest is the design from data of uncertainty sets for robust optimization. Constructing uncertainty sets from data is a central problem in robust optimization that has not been adequately addressed, and machine learning methodology may be able to provide a way forward.

14.8 References

N. Alon, S. Ben-David, N. Cesa-Bianchi, and D. Haussler. Scale-sensitive dimension, uniform convergence, and learnability. *Journal of the ACM*, 44(4):615–631, 1997.

M. Anthony and P. L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999.

P. L. Bartlett. The sample complexity of pattern classification with neural networks: The size of the weight is more important than the size of the network. *IEEE Transactions on Information Theory*, 44(2):525–536, 1998.

P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, November 2002.

P. L. Bartlett, O. Bousquet, and S. Mendelson. Local Rademacher complexities. *Annals of Statistics*, 33(4):1497–1537, 2005.

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

A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton University Press, 2009.

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

D. Bertsimas, D. B. Brown, and C. Caramanis. Theory and applications of robust optimization. To appear in SIAM Review, 2010.

C. Bhattacharyya, L. R. Grate, M. I. Jordan, L. El Ghaoui, and I. S. Mian. Robust sparse hyperplane classifiers: Application to uncertain molecular profiling data. *Journal of Computational Biology*, 11(6):1073–1089, 2004.

P. Bickel, Y. Ritov, and A. Tsybakov. Simultaneous analysis of Lasso and Dantzig selector. *Annals of Statistics*, 37(4):1705–1732, 2009.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152, New York, 1992. ACM Press.

O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

E. J. Candès and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52 (12):5406–5425, 2006.

E. J. Candès and T. Tao. The Dantzig selector: Statistical estimation when p is much larger than n. *Annals of Statistics*, 35(6):2313–2351, 2007.

E. J. Candès and T. Tao. Reflections on compressed sensing. *IEEE Information Theory Society Newsletter*, 58(4):20–23, 2008.

E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

C. Caramanis and S. Mannor. Learning in the limit with adversarial disturbances. In *Proceedings of the 21st Annual Conference on Learning Theory*, pages 467–478, 2008.

S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1999.

C. Cortes and V. N. Vapnik. Support vector networks. *Machine Learning*, 20:1–25, 1995.

A. d'Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *SIAM Review*, 49 (3):434–448, 2007.

L. Devroye and L. Györfi. *Nonparametric Density Estimation: the l_1 View*. John Wiley & Sons, 1985.

L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition.*

Springer, New York, 1996.

D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

G. E. Dullerud and F. Paganini. *A Course in Robust Control Theory: A Convex Approach*, volume 36 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 2000.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.

L. El Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, 18(4):1035–1064, 1997.

T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. In A. J. Smola, P. L. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 171–203, Cambridge, MA, 2000. MIT Press.

A. Globerson and S. Roweis. Nightmare at test time: Robust learning by feature deletion. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 353–360, New York, 2006. ACM Press.

A. Hoerl. Application of ridge analysis to regression problems. *Chemical Engineering Progress*, 58:54–59, 1962.

P. Kall and S. W. Wallace. *Stochastic Programming*. John Wiley & Sons, 1994.

S.-J. Kim, A. Magnani, and S. Boyd. Robust Fisher discriminant analysis. In *Advances in Neural Information Processing Systems*, pages 659–666, 2005.

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

S. Mukherjee, P. Niyogi, T. Poggio, and R. Rifkin. Learning theory: Stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization. *Advances in Computational Mathematics*, 25(1-3): 161–193, 2006.

A. Nilim and L. El Ghaoui. Robust control of Markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, September 2005.

E. Parzen. On the estimation of a probability density function and the mode. *Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

T. Poggio, R. Rifkin, S. Mukherjee, and P. Niyogi. General conditions for predictivity in learning theory. *Nature*, 428(6981):419–422, 2004.

A. Prékopa. *Stochastic Programming*. Kluwer Academic Publishers, 1995.

M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *Annals of Mathematical Statistics*, 27(3):832–837, 1956.

B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2001.

D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, New York, 1992.

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

A. L. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21(5):1154–1157, 1973.

I. Steinwart. Consistency of support vector machines and other regularized kernel classifiers. *IEEE Transactions on Information Theory*, 51(1):128–142, 2005.

I. Steinwart and A. Christmann. *Support Vector Machines*. Springer, New York, 2008.

R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.

A. N. Tikhonov and V. Arsenin. *Solutions of Ill-Posed Problems*. Winston, New York, 1977.

T. Trafalis and R. Gilbert. Robust support vector machines for classification and computational issues. *Optimization Methods and Software*, 22(1):187–198, February 2007.

A. W. van der Vaart and J. A. Wellner. *Weak Convergence and Empirical Processes*. Springer-Verlag, New York, 2000.

V. N. Vapnik and A. Chervonenkis. The necessary and sufficient conditions for consistency in the empirical risk minimization method. *Pattern Recognition and Image Analysis*, 1(3):284–305, 1991.

V. N. Vapnik and A. Lerner. Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24:744–780, 1963.

M. Wainwright. Sharp thresholds for noisy and high-dimensional recovery of sparsity using ℓ_1-constrained quadratic programming (Lasso). *IEEE Transactions on Information Theory*, 55:2183–2202, 2009.

H. Xu and S. Mannor. Robustness and generalization. ArXiv 1005.2243, 2010a.

H. Xu and S. Mannor. Robustness and generalizability. In *Proceeding of the 23rd Annual Conference on Learning Theory*, pages 503–515, 2010b.

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

H. Xu, C. Caramanis, and S. Mannor. Robust regression and Lasso. *IEEE Transactions on Information Theory*, 56(7):3561–3574, 2010a.

H. Xu, C. Caramanis, and S. Mannor. A distributional interpretation to robust optimization. submitted, 2010b.

H. Xu, C. Caramanis, and S. Sanghavi. Robust PCA via outlier pursuit. To appear *Advances in Neural Information Processing Systems*, 2010c.

K. Zhou, J. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice-Hall, 1996.