Mach Learn (2012) 86:391–423
DOI 10.1007/s10994-011-5268-1

Robustness and generalization

Huan Xu · Shie Mannor

Received: 16 February 2011 / Accepted: 10 October 2011 / Published online: 15 November 2011

Abstract We derive generalization bounds for learning algorithms based on their robustness: the property that if a testing sample is "similar" to a training sample, then the testing error is close to the training error. This provides a novel approach, different from complexity or stability arguments, to study generalization of learning algorithms. One advantage of the robustness approach, compared to previous methods, is the geometric intuition it conveys. Consequently, robustness-based analysis is easy to extend to learning in non-standard setups such as Markovian samples or quantile loss. We further show that a weak notion of robustness is both sufficient and necessary for generalizability, which implies that robustness is a fundamental property that is required for learning algorithms to work.

Keywords Generalization · Robust · Non-IID sample · Quantile loss

1 Introduction

The key issue in the task of learning from a set of observed samples is the estimation of the *risk* (i.e., generalization error) of learning algorithms. Typically, its empirical measurement (i.e., training error) provides an optimistically biased estimation, especially when the number of training samples is small. Several approaches have been proposed to bound the deviation of the risk from its empirical measurement, among which methods based on uniform convergence and stability are most widely used.

Uniform convergence of empirical quantities to their mean (Vapnik and Chervonenkis 1974, 1991) provides ways to bound the gap between the expected risk and the empirical risk by the complexity of the hypothesis set. Examples of complexity measures are

Editor: Phil Long.

H. Xu (✉)
Department of Mechanical Engineering, National University of Singapore, Singapore, Singapore
e-mail: mpexuh@nus.edu.sg

S. Mannor
Department of Electrical Engineering, Technion, Israel Institute of Technology, Haifa, Israel
e-mail: shie@ee.technion.ac.il

the Vapnik-Chervonenkis (VC) dimension (Vapnik and Chervonenkis 1991; Evgeniou et al. 2000), the fat-shattering dimension (Kearns and Schapire 1994; Alon et al. 1997; Bartlett 1998), and the Rademacher complexity (Koltchinskii 2002; Bartlett and Mendelson 2002; Bartlett et al. 2005). Another well-known approach is based on *stability*. An algorithm is stable if its output remains "similar" for different sets of training samples that are identical up to removal or change of a single sample. The first results that relate stability to generalizability track back to Devroye and Wagner (1979a, 1979b). Later, McDiarmid's concentration inequalities (McDiarmid 1989) facilitated new bounds on generalization error (e.g., Bousquet and Elisseeff 2002; Poggio et al. 2004; Mukherjee et al. 2006).

In this paper we explore a different approach which we term *algorithmic robustness*. Briefly speaking, an algorithm is robust if its solution has the following property: it achieves "similar" performance on a testing sample and a training sample that are "close". This notion of robustness is rooted in *robust optimization* (Ben-Tal and Nemirovski 1998, 1999; Bertsimas and Sim 2004) where a decision maker aims to find a solution x that minimizes a (parameterized) cost function $f(x,\xi)$ with the knowledge that the unknown true parameter ξ may deviate from the observed parameter $\hat{\xi}$. Hence, instead of solving $\min_x f(x,\hat{\xi})$ one solves $\min_x[\max_{\tilde{\xi}\in\Delta} f(x,\tilde{\xi})]$, where Δ includes all possible realizations of ξ. Robust optimization was introduced in machine learning tasks to handle exogenous noise (Bhattacharyya et al. 2004; Shivaswamy et al. 2006; Globerson and Roweis 2006), i.e., the learning algorithm only has access to inaccurate observation of training samples. Later on, Xu et al. (2009a, 2010b) showed that both Support Vector Machines (SVMs) and Lasso have robust optimization interpretation, i.e., they can be reformulated as

$$\min_{h\in\mathcal{H}} \max_{(\delta_1,\ldots,\delta_n)\in\Delta} \sum_{i=1}^{n} l(h, z_i+\delta_i),$$

for some Δ and $\mathcal{H}$. Here z_i are the observed training samples and $l(\cdot,\cdot)$ is the loss function (hinge-loss for SVMs, and squared loss for Lasso), which means that SVMs and Lasso essentially minimize the empirical error under the worst possible perturbation in some properly defined uncertainty set. Indeed, Xu et al. (2009a, 2010b) showed that this reformulation implies that the loss of a sample "close" to z_i is small, which further implies statistical consistency of these two algorithms. In this paper we adopt this approach and study the (finite sample) generalization ability of learning algorithms by investigating the loss of learned hypotheses on samples that slightly deviate from training samples.

We emphasize that one advantage of the proposed *algorithmic robustness* approach is that it is applicable to a very general setup. The standard setup in machine learning is restricted to the case that all samples are drawn in an IID fashion and the goal of learning is to minimize the *expected* loss (or error). Previous approaches critically depend on these assumptions. Extension, if possible, to non-standard setups—setups where either data are not IID or the minimizing objective is not the expected loss—often requires specifically tailored analysis (e.g. Gamarnik 2003; Lozano et al. 2006; Zou et al. 2009). In contrast, extension of robustness-based analysis to non-standard setups is straightforward. Indeed, we provide generalization bounds for two "non-standard" setups: one where samples are generated according to a Markovian chain, and one where the goal of learning is to minimize the quantile loss, using essentially same analysis as that of the standard setup. These setups arise naturally from reinforcement learning, time series and learning with outliers (Sutton and Barto 1998; Klivans et al. 2009).

Of special interest is that robustness is more than just another way to establish generalization bounds. Indeed, we show that a weaker notion of robustness is a *necessary and sufficient* condition of (asymptotic) generalizability of general learning algorithms. While it is

known that having a finite VC-dimension (Vapnik and Chervonenkis 1991) or equivalently being CVEEE$_{loo}$ stable (Mukherjee et al. 2006) is necessary and sufficient for Empirical Risk Minimization (ERM) to generalize, much less is known in the general case. Recently, Shalev-Shwartz et al. (2009) proposed a weaker notion of stability that is necessary and sufficient for a learning algorithm to be consistent and generalizing, provided that the problem itself is *learnable*. However, learnability requires that the *convergence rate is uniform* with respect to all distributions, and is hence a fairly strong assumption. In particular, the standard supervised learning setup where the hypothesis set is the set of measurable functions is *not* learnable since no algorithm can achieve a uniform convergence rate (Devroye et al. 1996). Indeed, as Shalev-Shwartz et al. (2009) stated, in the supervised learning setup, it is known that requiring the problem to be learnable itself is equivalent to requiring that the ERM algorithm generalizes. As aforementioned, the latter is only possible when the hypothesis set has finite VC dimensions.

In particular, our main contributions are the following:

1. We propose a notion of algorithmic robustness. Algorithmic robustness is a desired property for a learning algorithm since it implies a lack of sensitivity to (small) disturbances in the training data.
2. Based on the notion of algorithmic robustness, we derive generalization bounds for robust learning algorithms. Due to the geometric intuition the robust approach conveys, it is relatively easy to extend the analysis to non-standard setups—setups where the samples are not IID or the loss function is not the expected loss. In particular, we derived PAC bounds in the case where samples are drawn according to a Markovian chain, and in the case where the loss function is the quantile loss. This indicates that the fundamental nature of the proposed approach.
3. To illustrate the applicability of the notion of algorithmic robustness, we provide some examples of robust algorithms, including SVM, Lasso, feed-forward neural networks and PCA.
4. We propose a weaker notion of robustness and show that it is both necessary and sufficient for a learning algorithm to generalize. This implies that robustness is an essential property needed for a learning algorithm to work.

Note that while stability and robustness are similar on an intuitive level, there is a difference between the two: stability requires that nearly identical training sets with a single sample removed lead to similar prediction rules, whereas robustness requires that a prediction rule has comparable performance if tested on a sample close to a training sample.

We remark that in this paper we consider the relationship between robustness and generalizability. An equally important property of learning algorithms is *consistency*: the property that a learning algorithm guarantees to recover the global optimal solution as the number of training data increases. While it is straightforward that if an algorithm minimizes the empirical error asymptotically and also generalizes (or equivalently is *weakly robust*), then it is consistent, much less is known for a necessary condition for an algorithm to be consistent. It is certainly interesting to investigate the relationship between consistency and robustness, and in particular whether robustness is necessary for consistency, at least for algorithms that asymptotically minimize the empirical error.

A preliminary version of this paper has appeared in COLT 2010 (Xu and Mannor 2010). In the current version, we provide all proofs omitted in the conference version due to space constraints. More importantly, the current version extends the conference version in three directions. First, we discuss the relationship between robust optimization and generalization, which provides a method to construct learning algorithms with good generalization ability.

Second, we presents robustness-based generalization bounds for the case where samples are Markovian. Finally, we provide a detailed comparison of the proposed robustness-approach with existing approaches.

This paper is organized as follows. We define the notion of robustness in Sect. 2, and prove generalization bounds for robust algorithms in Sect. 3. In Sect. 5 we propose a relaxed notion of robustness, which is termed as pseudo-robustness, and show corresponding generalization bounds. Examples of learning algorithms that are robust or pseudo-robust are provided in Sect. 6. We further compare the proposed approach with previous approaches in Sect. 7. Finally, we show that robustness is necessary and sufficient for generalizability in Sect. 8.

1.1 Preliminaries

We consider the following general learning model: a set of training samples are given, and the goal is to pick a hypothesis from a hypothesis set. Unless otherwise mentioned, throughout this paper the size of training set is fixed as n. Therefore, we drop the dependence of parameters (that quantify the robustness of an algorithm) on the number of training samples, while it should be understood that these parameters may vary with the number of training samples. We use $\mathcal{Z}$ and $\mathcal{H}$ to denote the set from which each sample is drawn, and the hypothesis set, respectively. Throughout the paper we use $\mathbf{s}$ to denote the training sample set consists of n training samples $(s_1, \ldots, s_n)$. A learning algorithm $\mathcal{A}$ is thus a mapping from $\mathcal{Z}^n$ 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 a point $z \in \mathcal{Z}$, there is an associated loss $l(h, z)$. We ignore the issue of measurability and further assume that $l(h, z)$ is non-negative and upper-bounded uniformly by a scalar M.[1]

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 $\mathcal{Y}$, i.e., to predict the y-component given x-component. We hence use $\mathcal{A}_{\mathbf{s}}(x)$ to represent the prediction of $x \in \mathcal{X}$ if trained on $\mathbf{s}$. We call $\mathcal{X}$ the input space and $\mathcal{Y}$ the output space. The output space can either be $\mathcal{Y} = \{-1, +1\}$ for a classification problem, or $\mathcal{Y} = \mathbb{R}$ for a regression problem. 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. To simplify notations, for a scalar c, we use $[c]^+$ to represent its non-negative part, i.e., $[c]^+ \triangleq \max(0, c)$.

We recall the following standard notion of covering number (Kolmogorov and Tihomirov 2002).

Definition 1 For a metric space S, ρ and $T \subset S$ we say that $\hat{T} \subset S$ is an *ϵ-cover* of T, if $\forall t \in T, \exists \hat{t} \in \hat{T}$ such that $\rho(t, \hat{t}) \leq \epsilon$. The *$\epsilon$-covering number* of T is

$$\mathcal{N}(\epsilon, T, \rho) = \min\{|\hat{T}| : \hat{T} \text{ is an } \epsilon - \text{cover of } T\}.$$

2 Robustness of learning algorithms

Before providing a precise definition of what we mean by "robustness" of an algorithm, we provide a couple of motivating examples that share a common property: if a testing sample

[1] Note that if all samples are IID following a distribution μ, then we can replace the boundedness assumption by a weaker assumption of the existence an integrable envelop function, i.e., there exist $\bar{l}(\cdot) : \mathcal{Z} \mapsto \mathbb{R}$ such that $l(h, z) \leq \bar{l}(z)$ for all $h \in \mathcal{H}$ and $z \in \mathcal{Z}$, and that $\int_{\mathcal{Z}} \bar{l}(z)\mu(dz) < +\infty$.

and a training sample are close to each other, then their associated losses are also close, a property we will later formalize as "robustness."

We first consider large-margin classifiers: Let $\mathcal{X}$ be a subset of a metric space equipped with a metric ρ, and the loss function be $l(A_{\mathbf{s}}, z) = \mathbf{1}(A_{\mathbf{s}}(z^{(x)}) \neq z^{(y)})$. Fix $\gamma > 0$. An algorithm $\mathcal{A}$ has a margin γ on training set $\mathbf{s}$ if for $j = 1, \ldots, n$

$$\mathcal{A}_{\mathbf{s}}(x) = \mathcal{A}_{\mathbf{s}}(s_j^{(x)}); \quad \forall x : \rho(x, s_j^{(x)}) < \gamma.$$

That is, any training sample is at least γ away from the classification boundary.

Example 1 Fix $\gamma > 0$ and let $K = 2\mathcal{N}(\gamma/2, \mathcal{X}, \rho)$. If $\mathcal{A}_{\mathbf{s}}$ has a margin γ, then $\mathcal{Z}$ can be partitioned into K disjoint sets, denoted by $\{C_i\}_{i=1}^K$, such that if s_j and $z \in \mathcal{Z}$ belong to a same C_i, then $|l(\mathcal{A}_{\mathbf{s}}, s_j) - l(\mathcal{A}_{\mathbf{s}}, z)| = 0$.

Proof By definition of covering number, we can partition $\mathcal{X}$ into $\mathcal{N}(\gamma/2, \mathcal{X}, \rho)$ subsets (denoted $\hat{X}_i$) such that each subset has a diameter less or equal to γ. Further, $\mathcal{Y}$ can be partitioned to $\{-1\}$ and $\{+1\}$. Thus, we can partition $\mathcal{Z}$ into $2\mathcal{N}(\gamma/2, \mathcal{X}, \rho)$ subsets such that if z_1, z_2 belong to a same subset, then $z_1^{(y)} = z_2^{(y)}$ and $\rho(z_1^{(x)}, z_2^{(x)}) \leq \gamma$. By definition of margin, this guarantees that if s_j and $z \in \mathcal{Z}$ belong to a same C_i, then $|l(\mathcal{A}_{\mathbf{s}}, s_j) - l(\mathcal{A}_{\mathbf{s}}, z)| = 0$. □

The next example is a linear regression algorithm. Let the loss function be $l(A_{\mathbf{s}}, z) = |z^{(y)} - A_{\mathbf{s}}(z^{(x)})|$, and let $\mathcal{X}$ be a bounded subset of $\mathbb{R}^m$ and fix $c > 0$. The norm-constrained linear regression algorithm is

$$\mathcal{A}_{\mathbf{s}} = \arg \min_{w \in \mathbb{R}^m : \|w\|_2 \leq c} \sum_{i=1}^{n} |s_i^{(y)} - w^\top s_i^{(x)}|, \tag{1}$$

i.e., minimizing the empirical error among all linear classifiers whose norm is bounded.

Example 2 Fix $\epsilon > 0$ and put $K = \mathcal{N}(\epsilon/2, \mathcal{X}, \|\cdot\|_2) \times \mathcal{N}(\epsilon/2, \mathcal{Y}, |\cdot|)$. Consider the algorithm as in (1). The set $\mathcal{Z}$ can be partitioned into K disjoint sets, such that if s_j and $z \in \mathcal{Z}$ belong to a same C_i, then

$$|l(\mathcal{A}_{\mathbf{s}}, s_j) - l(\mathcal{A}_{\mathbf{s}}, z)| \leq (c+1)\epsilon.$$

Note that we can generalize this example to the case where $\mathcal{Z}$ is a compact subset of an arbitrary Hilbert space.

Proof Denote $\mathcal{A}_{\mathbf{s}}$ by w. Similarly to the previous example, we can partition $\mathcal{Z}$ to $\mathcal{N}(\epsilon/2, \mathcal{X}, \|\cdot\|_2) \times \mathcal{N}(\epsilon/2, \mathcal{Y}, |\cdot|)$ subsets, such that if z_1, z_2 belong to a same C_i, then $\|z_1^{(x)} - z_2^{(x)}\|_2 \leq \epsilon$, and $|z_1^{(y)} - z_2^{(y)}| \leq \epsilon$. Since $\|w\|_2 \leq c$, we have

$$\begin{aligned} |l(w, z_1) - l(w, z_2)| &= \left| |z_1^{(y)} - w^\top z_1^{(x)}| - |z_2^{(y)} - w^\top z_2^{(x)}| \right| \\ &\leq |(z_1^{(y)} - w^\top z_1^{(x)}) - (z_2^{(y)} - w^\top z_2^{(x)})| \\ &\leq |z_1^{(y)} - z_2^{(y)}| + \|w\|_2 \|z_1^{(x)} - z_2^{(x)}\|_2 \\ &\leq (1+c)\epsilon, \end{aligned}$$

whenever z_1, z_2 belong to a same C_i. □

The two motivating examples both share a property: we can partition the sample set into finite subsets, such that if a new sample falls into the same subset as a testing sample, then the loss of the former is close to the loss of the latter. We call an algorithm having this property "robust."

Definition 2 Algorithm $\mathcal{A}$ is $(K, \epsilon(\cdot))$ robust, for $K \in \mathbb{N}$ and $\epsilon(\cdot) : \mathcal{Z}^n \mapsto \mathbb{R}$, if $\mathcal{Z}$ can be partitioned into K disjoint sets, denoted by $\{C_i\}_{i=1}^K$, such that the following holds for all $\mathbf{s} \in \mathcal{Z}^n$:

$$\forall s \in \mathbf{s}, \forall z \in \mathcal{Z}, \forall i = 1, \ldots, K : \text{ if } s, z \in C_i, \text{ then } |l(\mathcal{A}_\mathbf{s}, s) - l(\mathcal{A}_\mathbf{s}, z)| \leq \epsilon(\mathbf{s}). \quad (2)$$

The parameters K and $\epsilon(\cdot)$ quantify the robustness of an algorithm. Since $\epsilon(\cdot)$ is a function of training samples, for different training samples an algorithm may exhibit different robustness property. For example, a classification algorithm is more robust to a training set with a larger margin. Because (2) involves both the trained solution $\mathcal{A}_\mathbf{s}$ and the training set $\mathbf{s}$, robustness is a property of the learning algorithm, rather than the property of the "effective hypothesis space," i.e., all the hypotheses that can be output by the algorithm.

Note that the definition of robustness requires that (2) holds *for every* training sample. Indeed, we can relax the definition, so that the condition needs only hold for a subset of training samples. We call an algorithm having this property "pseudo robust." See Sect. 5 for details.

3 Generalization of robust algorithms: the standard setup

We now investigate generalization property of robust algorithms, by establishing PAC bounds for different setups. This section is devoted to the standard learning setup, i.e., the sample set $\mathbf{s}$ consists of n i.i.d. samples generated by an unknown distribution μ, and the goal of learning is to minimize expected test loss. Let $\mathcal{L}(\cdot)$ and $l_{\text{emp}}(\cdot)$ denote the expected error and the training error, i.e.,

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

Recall that the loss function $l(\cdot, \cdot)$ is upper bounded by M.

Theorem 1 *If a learning algorithm $\mathcal{A}$ is $(K, \epsilon(\cdot))$-robust, and the training sample set $\mathbf{s}$ is generated by n IID draws from μ, then for any $\delta > 0$, with probability at least $1 - \delta$ we have*

$$|\mathcal{L}(\mathcal{A}_\mathbf{s}) - l_{\text{emp}}(\mathcal{A}_\mathbf{s})| \leq \epsilon(\mathbf{s}) + M\sqrt{\frac{2K \ln 2 + 2\ln(1/\delta)}{n}}.$$

Proof Let N_i be the set of index of points of $\mathbf{s}$ that fall into the C_i. Note that $(|N_1|, \ldots, |N_K|)$ is an IID multinomial random variable with parameters n and $(\mu(C_1), \ldots, \mu(C_K))$. The following holds by the Breteganolle-Huber-Carol inequality (see Proposition A6.6 of van der Vaart and Wellner 2000):

$$\Pr\left\{\sum_{i=1}^K \left|\frac{|N_i|}{n} - \mu(C_i)\right| \geq \lambda\right\} \leq 2^K \exp\left(\frac{-n\lambda^2}{2}\right).$$

Hence, the following holds with probability at least $1-\delta$,

$$\sum_{i=1}^{K}\left|\frac{|N_i|}{n}-\mu(C_i)\right|\leq\sqrt{\frac{2K\ln 2+2\ln(1/\delta)}{n}}. \tag{3}$$

We have

$$\begin{aligned}
&|\mathcal{L}(\mathcal{A}_{\mathbf{s}})-l_{\mathrm{emp}}(\mathcal{A}_{\mathbf{s}})|\\
&=\left|\sum_{i=1}^{K}\mathbb{E}_{z\sim\mu}(l(\mathcal{A}_{\mathbf{s}},z)|z\in C_i)\mu(C_i)-\frac{1}{n}\sum_{i=1}^{n}l(\mathcal{A}_{\mathbf{s}},s_i)\right|\\
&\overset{(a)}{\leq}\left|\sum_{i=1}^{K}\mathbb{E}_{z\sim\mu}(l(\mathcal{A}_{\mathbf{s}},z)|z\in C_i)\frac{|N_i|}{n}-\frac{1}{n}\sum_{i=1}^{n}l(\mathcal{A}_{\mathbf{s}},s_i)\right|\\
&\quad+\left|\sum_{i=1}^{K}\mathbb{E}_{z\sim\mu}(l(\mathcal{A}_{\mathbf{s}},z)|z\in C_i)\mu(C_i)-\sum_{i=1}^{K}\mathbb{E}_{z\sim\mu}(l(\mathcal{A}_{\mathbf{s}},z)|z\in C_i)\frac{|N_i|}{n}\right|\\
&\overset{(b)}{\leq}\left|\frac{1}{n}\sum_{i=1}^{K}\sum_{j\in N_i}\max_{z_2\in C_i}|l(\mathcal{A}_{\mathbf{s}},s_j)-l(\mathcal{A}_{\mathbf{s}},z_2)|\right|+\left|\max_{z\in\mathcal{Z}}|l(\mathcal{A}_{\mathbf{s},z})|\sum_{i=1}^{K}\left|\frac{|N_i|}{n}-\mu(C_i)\right|\right|\\
&\overset{(c)}{\leq}\epsilon(\mathbf{s})+M\sum_{i=1}^{K}\left|\frac{|N_i|}{n}-\mu(C_i)\right|,
\end{aligned} \tag{4}$$

where (a), (b), and (c) are due to the triangle inequality, the definition of N_i, and the definition of $\epsilon(\mathbf{s})$ and M, respectively. Note that the right-hand-side of (4) is upper-bounded by $\epsilon(\mathbf{s})+M\sqrt{\frac{2K\ln 2+2\ln(1/\delta)}{n}}$ with probability at least $1-\delta$ due to (3). The theorem follows. □

Theorem 1 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 1 *If a learning algorithm $\mathcal{A}$ is $(K,\epsilon_K(\cdot))$-robust for all $K\geq 1$, and the training sample $\mathbf{s}$ is generated by n IID draws from μ, then for any $\delta>0$, with probability at least $1-\delta$,*

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

Proof Let

$$E(K)\triangleq\left\{|\mathcal{L}(\mathcal{A}_{\mathbf{s}})-l_{\mathrm{emp}}(\mathcal{A}_{\mathbf{s}})|>\epsilon_K(\mathbf{s})+M\sqrt{\frac{2K\ln 2+2\ln\frac{K(K+1)}{\delta}}{n}}\right\}.$$

From Theorem 1 we have $\Pr(E(K)) \leq \delta/(K(K+1)) = \delta/K - \delta/(K+1)$. By the union bound we have

$$\Pr\left\{\bigcup_{K\geq 1} E(K)\right\} \leq \sum_{K\geq 1}\Pr(E(K)) \leq \sum_{K\geq 1}\left[\frac{\delta}{K} - \frac{\delta}{K+1}\right] = \delta,$$

and the corollary follows. □

If $\epsilon(\cdot)$ is a constant, i.e., $\epsilon_K(\mathbf{s}) \triangleq \epsilon_K$ for all $\mathbf{s}$, then we can sharpen the bound given in Corollary 1.

Corollary 2 *If a learning algorithm $\mathcal{A}$ is (K, ϵ_K)-robust for all $K \geq 1$, and the training sample $\mathbf{s}$ is generated by n IID draws from μ, then for any $\delta > 0$, with probability at least $1-\delta$,*

$$|\mathcal{L}(\mathcal{A}_\mathbf{s}) - l_{\text{emp}}(\mathcal{A}_\mathbf{s})| \leq \inf_{K\geq 1}\left[\epsilon_K + M\sqrt{\frac{2K\ln 2 + 2\ln\frac{1}{\delta}}{n}}\right].$$

Proof The right hand side does not depend on $\mathbf{s}$, and hence the optimal K^*. Therefore, plugging K^* into Theorem 1 establishes the corollary. □

Let us comment on the dependence of K and ϵ on the training set, particularly on the number of training samples n. As we remarked in Sect. 1.1, we drop the dependence of the parameters on n because we are interested in finite-sample bounds as opposed to asymptotic rates, and because results in this section are general, and hold for all robust algorithms. However, the dependence of the parameters on n becomes explicit when studying individual robust algorithms (see Sect. 6 for detail). Specifically, combining results from Sect. 6 and the main theorems presented in this section, it is straight-forward to derive both finite-sample bounds and asymptotic rates for individual robust algorithms; see Sect. 7 for an example.

Generalization and robust optimization

Following a similar line as the proof of Theorem 1, one can easily show the following result.

Corollary 3 *Let $C_1, \ldots, C_K$ be a partition of $\mathcal{Z}$, and write $z_1 \sim z_2$ if z_1, z_2 fall into the same C_k. If the training sample $\mathbf{s}$ is generated by n IID draws from μ, then for any $\delta > 0$, with probability at least $1-\delta$, the following holds uniformly over $h \in \mathcal{H}$*

$$l_{\text{emp}}(h) \leq \frac{1}{n}\sum_{i=1}^{n}\max_{\hat{s}_i \sim s_i} l(h, \hat{s}_i) + M\sqrt{\frac{2K\ln 2 + 2\ln(1/\delta)}{n}}.$$

Corollary 3 suggests that one can use robust optimization to construct learning algorithms. Note that to make the empirical error small, we can minimize its upper bound—the right hand side, i.e., to solve the following robust optimization problem:

$$\operatorname*{Minimize}_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n}\max_{\hat{s}_i \sim s_i} l(h, \hat{s}_i).$$

In the recent years, robust optimization has been extensively used in machine learning (Bhattacharyya et al. 2004; Globerson and Roweis 2006; Lanckriet et al. 2003; Shivaswamy et al. 2006; Xu et al. 2009a, 2009b, 2010b, and many others); see Caramanis et al. (2011) for a comprehensive survey. It was observed that robust optimization based algorithms not only are robust to observation noise or disturbance, and often exhibit desirable generalization properties, as shown by Corollary 3.

4 Generalization of robust algorithms: non-standard setups

In this section we derive PAC bounds for robust algorithms under learning setups that are less extensively investigated, which includes: (1) The learning goal is to minimize quantile loss. (2) The samples are generated according to a (Doeblin) Markovian chain. Indeed, the fact that we can provide results in non-standard learning setups indicates the fundamental nature of robustness as a property of learning algorithms.

4.1 Quantile loss

As opposed to the standard expected loss setup, we consider some less extensively investigated loss functions, namely quantile value and truncated expectation (see below for precise definitions). These loss functions are of interest, and have been applied in many areas including ecology (Cade and Noon 2003), medicine (Cole 1988) and finance (Koenker and Bassett 1978), because they are less sensitive to the presence of outliers than the standard average loss (Huber 1981).

Learning from samples with outliers has attracted increasing attention in the recent decade (Klivans et al. 2009; Xu et al. 2010a; Yu et al. 2011, and many others). When a sample set contains a non-negligible fraction of data corrupted in an arbitrary or even adversary manner, the average or expected loss ceases to be a good measurement of the desirability of a solution. Instead, quantile measurements such as the median loss become more appropriate in this setup. However, generalization w.r.t. loss functions different than the expected loss is largely unexplored, partly due to the fact that classical approaches heavily rely on techniques (e.g., symmetrization, see Bartlett and Mendelson 2002; Bousquet et al. 2005; van der Vaart and Wellner 2000 for examples) that are built for, and hard to extend beyond, the expected loss case.

Definition 3 For a non-negative random variable X, the *β-quantile value* is

$$\mathbb{Q}^{\beta}(X) \triangleq \inf\{c \in \mathbb{R} : \Pr(X \leq c) \geq \beta\}.$$

The *β-truncated mean* is

$$\mathbb{T}^{\beta}(X) \triangleq \begin{cases} \mathbb{E}[X \cdot \mathbf{1}(X < \mathbb{Q}^{\beta}(X))] & \text{if } \Pr[X = \mathbb{Q}^{\beta}(X)] = 0; \\ \mathbb{E}[X \cdot \mathbf{1}(X < \mathbb{Q}^{\beta}(X))] + \frac{\beta - \Pr[X < \mathbb{Q}^{\beta}(X)]}{\Pr[X = \mathbb{Q}^{\beta}(X)]} \mathbb{Q}^{\beta}(X) & \text{otherwise.} \end{cases}$$

In words, the β-quantile loss is the smallest value that is larger or equal to X with probability at least β. The β-truncated mean is the contribution to the expectation of the leftmost β fraction of the distribution. For example, suppose X is supported on $\{c_1, \ldots, c_{10}\}$ ($c_1 < c_2 < \cdots < c_{10}$) and the probability of taking each value equals 0.1. Then the 0.63-quantile loss of X is c_7, and the 0.63-truncated mean of X equals $0.1(\sum_{i=1}^{6} c_i + 0.3c_7)$.

Given $h \in \mathcal{H}$, $\beta \in (0,\ 1)$, and a probability measure μ on $\mathcal{Z}$, let

$$\mathcal{Q}(h, \beta, \mu) \triangleq \mathbb{Q}^{\beta}(l(h, z)); \quad \text{where: } z \sim \mu;$$

and

$$\mathcal{T}(h, \beta, \mu) \triangleq \mathbb{T}^{\beta}(l(h, z)); \quad \text{where: } z \sim \mu;$$

i.e., the β-quantile value and β-truncated mean of the (random) testing error of hypothesis h if the testing sample follows distribution μ. We have the following theorem that is a special case of Theorem 5 below, hence we omit the proof.

Theorem 2 (Quantile value & truncated mean) *Suppose the training sample* $\mathbf{s}$ *is generated by* n *IID draws from* μ, *and denote the empirical distribution of* $\mathbf{s}$ *by* μ_{emp}. *Let* $\lambda_0 = \sqrt{\frac{2K \ln 2 + 2\ln(1/\delta)}{n}}$. *If* $0 \le \beta - \lambda_0 \le \beta + \lambda_0 \le 1$ *and* $\mathcal{A}$ *is* $(K, \epsilon(\cdot))$ *robust, then with probability at least* $1 - \delta$, *the followings hold*

(I) $\mathcal{Q}(\mathcal{A}_{\mathbf{s}}, \beta - \lambda_0, \mu_{\text{emp}}) - \epsilon(\mathbf{s}) \le \mathcal{Q}(\mathcal{A}_{\mathbf{s}}, \beta, \mu) \le \mathcal{Q}(\mathcal{A}_{\mathbf{s}}, \beta + \lambda_0, \mu_{\text{emp}}) + \epsilon(\mathbf{s})$;
(II) $\mathcal{T}(\mathcal{A}_{\mathbf{s}}, \beta - \lambda_0, \mu_{\text{emp}}) - \epsilon(\mathbf{s}) \le \mathcal{T}(\mathcal{A}_{\mathbf{s}}, \beta, \mu) \le \mathcal{T}(\mathcal{A}_{\mathbf{s}}, \beta + \lambda_0, \mu_{\text{emp}}) + \epsilon(\mathbf{s})$.

In words, Theorem 2 essentially means that with high probability, the β-quantile value/truncated mean of the testing error (recall that the testing error is a random variable) is (approximately) bounded by the $(\beta \pm \lambda_0)$-quantile value/truncated mean of the empirical error, thus providing a way to estimate the quantile value/truncated expectation of the testing error based on empirical observations.

4.2 Markovian samples

The robustness approach is not restricted to the IID setup. In many applications of interest, such as reinforcement learning and time series forecasting, the IID assumption is violated. In such applications there is a time driven process that generates samples that depend on the previous samples (e.g., the observations of a trajectory of a robot). Such a situation can be modeled by stochastic process such as a Markov process. In this section we establish a result similar to the IID case for samples that are drawn from a Markov chain. Such setup has been proposed in Gamarnik (2003) for a finite and countable state space in the context of additive loss. Instead, we consider the case where the *state space can be general*, i.e., it is not necessarily finite or countable. Thus, a certain ergodic structure of the underlying Markov chain is needed. We focus on chains that converge to equilibrium exponentially fast and uniformly in the initial condition. It is known that this is equivalent to the class of Doeblin chains (Meyn and Tweedie 1993). Thus, it easy to see that all finite Markovian chains are Doeblin Chains. Recall the following definition (Meyn and Tweedie 1993; Doob 1953).

Definition 4 A Markov chain $\{z_i\}_{i=1}^{\infty}$ on a state space $\mathcal{Z}$ is a *Doeblin chain* (with α and T) if there exists a probability measure φ on $\mathcal{Z}$, $\alpha > 0$, an integer $T \ge 1$ such that

$$\Pr(z_T \in H | z_0 = z) \ge \alpha \varphi(H); \quad \forall \text{ measurable } H \subseteq \mathcal{Z};\ \forall z \in \mathcal{Z}.$$

The class of Doeblin chains is probably the "nicest" class of general state-space Markov chains. We notice that such assumption is not overly restrictive, since by requiring that an

ergodic theorem holds for all bounded functions uniformly in the initial distribution itself implies that a chain is Doeblin (Meyn and Tweedie 1993). In particular, an ergodic chain defined on a finite state-space is a Doeblin chain.

Indeed, the Doeblin chain condition guarantees that an invariant measure π exists. Furthermore, we have the following lemma adapted from Theorem 2 of Glynn and Ormoneit (2002).

Lemma 1 *Let $\{z_i\}$ be a Doeblin chain as in Definition 4. Fix a function $f : \mathcal{Z} \to \mathbb{R}$ such that $\|f\|_\infty \leq C$. Then for $n > 2CT/\epsilon\alpha$ the following holds*

$$\Pr\left(\frac{1}{n}\sum_{i=1}^{n} f(z_i) - \int_{\mathcal{Z}} f(z)\pi(dz)s \geq \epsilon\right) \leq \exp\left(-\frac{\alpha^2(n\epsilon - 2CT/\alpha)^2}{2nC^2T^2}\right).$$

The following is the main theorem of this section that establishes a generalization bound for robust algorithms with samples drawn according to a Doeblin chain.

Theorem 3 *Suppose $\mathcal{A}$ is $(K, \epsilon(\cdot))$-robust. If $\mathbf{s} = \{s_1, \ldots, s_n\}$ is generated as the first n outputs of a Doeblin chain with α and T such that $n > 2T/\alpha$, then for any $\delta > 0$, with probability at least $1 - \delta$,*

$$|\mathcal{L}(\mathcal{A}_\mathbf{s}) - l_{\text{emp}}(\mathcal{A}_\mathbf{s})| \leq \epsilon(\mathbf{s}) + M\left\{\frac{2T}{\alpha n} + \sqrt{\frac{2T^2(K\ln 2 + \ln(1/\delta))}{\alpha^2 n}}\right\}.$$

The proof of Theorem 3 closely resembles that of Theorem 1, with some additional effort to handle the fact that training samples are not IID. We hence defer the proof to Appendix A.

5 Pseudo robustness

In this section we propose a relaxed definition of robustness that accounts for the case where (2) holds for most of training samples, as opposed to Definition 3 where (2) holds for all training samples. Recall that the size of training set is fixed as n.

Definition 5 Algorithm $\mathcal{A}$ is $(K, \epsilon(\cdot), \hat{n}(\cdot))$ *pseudo robust*, for $K \in \mathbb{N}$, $\epsilon(\cdot) : \mathcal{Z}^n \mapsto \mathbb{R}$ and $\hat{n}(\cdot) : \mathcal{Z}^n \mapsto \{1, \ldots, n\}$, if $\mathcal{Z}$ can be partitioned into K disjoint sets, denoted as $\{C_i\}_{i=1}^K$, such that for all $\mathbf{s} \in \mathcal{Z}^n$, there exists a subset of training samples $\hat{\mathbf{s}}$ with $|\hat{\mathbf{s}}| = \hat{n}(\mathbf{s})$ that the following holds:

$$\forall s \in \hat{\mathbf{s}},\ \forall z \in \mathcal{Z}, \forall i = 1, \ldots, K:\ \text{if } s, z \in C_i,\ \text{then } |l(\mathcal{A}_\mathbf{s}, s) - l(\mathcal{A}_\mathbf{s}, z)| \leq \epsilon(\mathbf{s}).$$

Observe that $(K, \epsilon(\cdot))$-robust is equivalent to $(K, \epsilon(\cdot), n)$ pseudo robust.

Theorem 1 can be generalized to the pseudo robust case. We defer the detailed proof to Appendix B.

Theorem 4 *If a learning algorithm $\mathcal{A}$ is $(K, \epsilon(\cdot), \hat{n}(\cdot))$ pseudo robust, and the training sample set $\mathbf{s}$ is generated by n IID draws from μ, then for any $\delta > 0$, with probability at least $1 - \delta$ we have*

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

Similarly, Theorem 2 can be generalized to the pseudo robust case. The proof is lengthy and hence postponed to Appendix C.

Theorem 5 (Quantile value & truncated expectation) *Suppose* $\mathbf{s}$ *has* n *samples drawn i.i.d. according to* μ, *and denote the empirical distribution of* $\mathbf{s}$ *by* μ_{emp}. *Let* $\lambda_0 = \sqrt{\frac{2K\ln 2 + 2\ln(1/\delta)}{n}}$. *Suppose that* $0 \le \beta - \lambda_0 - (n-\hat{n})/n \le \beta + \lambda_0 + (n-\hat{n})/n \le 1$ *and* $\mathcal{A}$ *is* $(K, \epsilon(\cdot), \hat{n}(\cdot))$ *pseudo robust. Then with probability at least* $1-\delta$, *the following holds*

$$\begin{aligned}
\text{(I)}\quad & \mathcal{Q}\left(\mathcal{A}_{\mathbf{s}}, \beta - \lambda_0 - \frac{n-\hat{n}(\mathbf{s})}{n}, \mu_{\text{emp}}\right) - \epsilon(\mathbf{s}) \\
& \le \mathcal{Q}(\mathcal{A}_{\mathbf{s}}, \beta, \mu) \le \mathcal{Q}\left(\mathcal{A}_{\mathbf{s}}, \beta + \lambda_0 + \frac{n-\hat{n}(\mathbf{s})}{n}, \mu_{\text{emp}}\right) + \epsilon(\mathbf{s}); \\
\text{(II)}\quad & \mathcal{T}\left(\mathcal{A}_{\mathbf{s}}, \beta - \lambda_0 - \frac{n-\hat{n}(\mathbf{s})}{n}, \mu_{\text{emp}}\right) - \epsilon(\mathbf{s}) \\
& \le \mathcal{T}(\mathcal{A}_{\mathbf{s}}, \beta, \mu) \le \mathcal{T}\left(\mathcal{A}_{\mathbf{s}}, \beta + \lambda_0 + \frac{n-\hat{n}(\mathbf{s})}{n}, \mu_{\text{emp}}\right) + \epsilon(\mathbf{s}).
\end{aligned}$$

Generalizing the concept of pseudo robustness to the Markovian setup is straightforward, and hence we omit the details.

6 Examples of robust algorithms

In this section we provide some examples of robust algorithms. The proofs of the examples can be found in Appendices D–I. Our first example is Majority Voting (MV) classification (see Sect. 6.3 of Devroye et al. 1996) that partitions the input space $\mathcal{X}$ and labels each partition set according to a majority vote of the training samples belonging to it.

Example 3 (Majority voting) Let $\mathcal{Y} = \{-1, +1\}$. Partition $\mathcal{X}$ to $\mathcal{C}_1, \ldots, \mathcal{C}_K$, and use $\mathcal{C}(x)$ to denote the set to which x belongs. A new sample $x_a \in \mathcal{X}$ is labeled by

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

If the loss function is $l(\mathcal{A}_s, z) = f(z^{(y)}, \mathcal{A}_{\mathbf{s}}(z^{(x)}))$ for some function f, then MV is $(2K, 0)$ robust.

MV algorithm has a natural partition of the sample space that makes it robust. Another class of robust algorithms are 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. The next theorem states that an algorithm is robust if two samples being close implies that they have similar testing error.

Theorem 6 *Fix* $\gamma > 0$ *and metric* ρ *of* $\mathcal{Z}$. *Suppose* $\mathcal{A}$ *satisfies*

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

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

Proof Let $\{c_1, \ldots, c_{\mathcal{N}(\gamma/2,\mathcal{Z},\rho)}\}$ be a $\gamma/2$-cover of $\mathcal{Z}$. whose existence is guaranteed by the definition of covering number. Let $\hat{C}_i = \{z \in \mathcal{Z} | \rho(z, c_i) \leq \gamma/2\}$, and $C_i = \hat{C}_i \cap (\bigcup_{j=1}^{i-1} \hat{C}_j)^c$. Thus, $C_1, \ldots, C_{\mathcal{N}(\gamma/2,\mathcal{Z},\rho)}$ is a partition of $\mathcal{Z}$, and satisfies

$$z_1, z_2 \in C_i \quad \Longrightarrow \quad \rho(z_1, z_2) \leq \rho(z_1, c_i) + \rho(z_2, c_i) \leq \gamma.$$

Therefore,

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

implies

$$z_1 \in \mathbf{s}\ z_1, z_2 \in C_i \quad \Longrightarrow \quad |l(\mathcal{A}_{\mathbf{s}}, z_1) - l(\mathcal{A}_{\mathbf{s}}, z_2)| \leq \epsilon(\mathbf{s}),$$

and the theorem follows. □

Theorem 6 immediately leads to the next example: if the testing error given the output of an algorithm is Lipschitz continuous, then the algorithm is robust.

Example 4 (Lipschitz continuous functions) If $\mathcal{Z}$ is compact w.r.t. metric ρ, $l(\mathcal{A}_{\mathbf{s}}, \cdot)$ is Lipschitz continuous with Lipschitz constant $c(\mathbf{s})$, i.e.,

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

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

Theorem 6 also implies that SVM, Lasso, feed-forward neural network and PCA are robust, as stated in Examples 5 to 8. The proofs are deferred to Appendix E to H.

Example 5 (Support vector machine) Let $\mathcal{X}$ be compact. Consider the standard SVM formulation (Cortes and Vapnik 1995; Schölkopf and Smola 2002)

$$\begin{aligned} \underset{w,d,\xi_1,\ldots,\xi_n}{\text{Minimize}}\ & c\|w\|_{\mathbb{H}}^2 + \frac{1}{n}\sum_{i=1}^{n}\xi_i \\ \text{s.t.}\quad & 1 - s_i^{(y)}[\langle w,\ \phi(s_i^{(x)})\rangle + d] \leq \xi_i; \\ & \xi_i \geq 0. \end{aligned}$$

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

[2]More precisely, let $\mathbb{H}$ be a Hilbert space, equipped with an inner product operator $\langle\cdot,\cdot\rangle$. A feature mapping $\phi(\cdot)$ is a continuous mapping from $\mathcal{X}$ to $\mathbb{H}$. The norm $\|\cdot\|_{\mathbb{H}} : \mathbb{H} \mapsto \mathbb{R}$ is defined as $\|w\|_{\mathbb{H}} = \langle w, w\rangle$, for all $w \in \mathbb{H}$. The kernel function $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is defined as $k(x_1, x_2) = \langle\phi(x_1), \phi(x_2)\rangle$.

Example 6 (Lasso) Let $\mathcal{Z}$ be a compact subset of $\mathbb{R}^{m+1}$, 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:

$$\min_{w} \frac{1}{n}\sum_{i=1}^{n}(s_i^{(y)} - w^{\top}s_i^{(x)})^2 + c\|w\|_1, \tag{5}$$

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

A close examination of the robustness result of SVM and Lasso shows that the resulting generalization bound is not tight compared to standard methods such as VC-dimension based results. More specifically, robustness based results depend exponentially on the dimension of $\mathcal{Z}$, while the VC-dimension based results depend linearly on the dimension of $\mathcal{Z}$. The reason is that the solution of SVM or Lasso belongs to the set of linear decision boundaries—a rather restrictive hypothesis set that in particular has a small VC dimensionality—leading to a favorable bound using VC-dimension approach. However, the robustness based approach only exploits the Liptschitz continuity, but not the linearity of these algorithms, which results in an inferior result. To close this gap, we suspect that an adaptive partition scheme may help. See Sect. 9 for a detailed discussion.

Example 7 (Feed-forward neural networks) Let $\mathcal{Z}$ be a compact subset of $\mathbb{R}^{m+1}$ 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, which is the following predicting rule given an input $x \in \mathcal{X}$:

$$\begin{aligned} x^0 &:= z^{(x)} \\ \forall v = 1, \ldots, d-1: \quad x_i^v &:= \sigma\left(\sum_{j=1}^{N_{v-1}} w_{ij}^{v-1}(\mathbf{s}) \cdot x_j^{v-1}\right); \quad i = 1, \ldots, N_v; \\ \mathcal{A}_{\mathbf{s}}(x) &:= \sigma\left(\sum_{j=1}^{N_{d-1}} w_j^{d-1}(\mathbf{s}) \cdot x_j^{d-1}\right), \end{aligned}$$

where weights $w_{ij}^v(\cdot)$ are trained using a learning algorithm $\mathcal{A}$ (e.g., backward propagation). Define $\alpha(\cdot) : \mathcal{Z}^n \mapsto \mathbb{R}$ and constant β as,

$$\alpha(\mathbf{s}) \triangleq \max_{v\in[1:d]\, i\in[1:N_v]} \sum_{j=1}^{N_{v-1}} |w_{ij}^{v-1}(\mathbf{s})|, \quad \forall \mathbf{s} \in \mathcal{Z}^n; \qquad \beta \triangleq \max_{a,b\in\mathbb{R}, a\neq b} \frac{|\sigma(a) - \sigma(b)|}{|a-b|},$$

then $\mathcal{A}$ is $(\mathcal{N}(\gamma/2, \mathcal{Z}, \|\cdot\|_\infty), \alpha(\cdot)^d\beta^d\gamma)$-robust, for all $\gamma > 0$.

Remark 1 In the classification case, where $\mathcal{Y} = \{-1, +1\}$, the real-valued output prediction $\mathcal{A}_{\mathbf{s}}(x)$ can be convert into a binary output $\hat{\mathcal{A}}_{\mathbf{s}}(x)$ via

$$\hat{\mathcal{A}}_{\mathbf{s}}(x) = \text{sign}(\mathcal{A}_{\mathbf{s}}(x)).$$

Observe that for all $v \in \mathbb{R}$ and $w \in \{-1, +1\}$, one has that $\mathbf{1}(\text{sign}(v) \neq w) \leq |v - w|$. Hence, since $y \in \{-1, +1\}$, we can upper-bound the expected classification error using $\mathbb{E}_{(x,y)\sim\mu}\mathbf{1}(\hat{\mathcal{A}}_{\mathbf{s}}(x)) \neq y) \leq \mathcal{L}(\mathcal{A}_{\mathbf{s}})$.

It is worthwhile noticing that in Example 7, the number of hidden units in each layer has no effect on the robustness of the algorithm and consequently 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.

The next example considers an unsupervised learning algorithm, namely the principal component analysis. We show that it is robust if the sample space is *bounded*. Note that, this does not contradict with the well known fact that the principal component analysis is sensitive to outliers which are far away from the origin.

Example 8 (Principal component analysis (PCA)) Let $\mathcal{Z} \subset \mathbb{R}^m$, such that $\max_{z\in\mathcal{Z}} \|z\|_2 \leq B$. If the loss function is $l((w_1,\ldots,w_d),z) = \sum_{k=1}^{d}(w_k^\top z)^2$, then finding the first d principal components, which solves the following optimization problem of $w_1,\ldots,w_d \in \mathbb{R}^m$,

$$\begin{aligned} \text{Maximize } & \sum_{i=1}^{n}\sum_{k=1}^{d}(w_k^\top s_i)^2 \\ \text{s.t. } & \|w_k\|_2 = 1,\ k = 1,\ldots,d; \\ & w_i^\top w_j = 0,\ i \neq j. \end{aligned}$$

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

The last example is large-margin classification, which is a generalization of Example 1. We need the following standard definition (e.g., Bartlett 1998) of the distance of a point to a classification rule.

Definition 6 Fix a metric ρ of $\mathcal{X}$. Given a classification rule Δ and $x \in \mathcal{X}$, the *distance* of x to Δ is

$$\mathcal{D}(x,\Delta) \triangleq \inf\{c \geq 0 | \exists x' \in \mathcal{X} : \rho(x,x') \leq c, \Delta(x) \neq \Delta(x')\}.$$

A large margin classifier is a classification rule such that most of the training samples are "far away" from the classification boundary. More precisely, the following example quantifies the robustness of an arbitrary classification based on its margin.

Example 9 (Large-margin classifier) Let $\gamma > 0$. Given a classification algorithm $\mathcal{A}$, define $\hat{n} : \mathcal{Z}^n \mapsto \mathbb{R}$ as

$$\hat{n}(\mathbf{s}) \triangleq \sum_{i=1}^{n} \mathbf{1}(\mathcal{D}(s_i^{(x)}, \mathcal{A}_s) > \gamma), \quad \forall \mathbf{s} \in \mathcal{Z}^n,$$

then algorithm $\mathcal{A}$ is $(2\mathcal{N}(\gamma/2,\mathcal{X},\rho), 0, \hat{n}(\cdot))$ pseudo robust, provided that $\mathcal{N}(\gamma/2,\mathcal{X},\rho) < \infty$.

Remark 2 In all examples except the first one, we assume that the sample space $\mathcal{Z}$ is compact, so that it can be covered by a finite number of subsets with bounded diameters. Note that as in previous works (e.g., Steinwart 2006), this assumption is mainly introduced to simplify the exposition. Indeed, to extend our analysis to a non-compact $\mathcal{Z}$, for any $\eta > 0$ we

can pick $\hat{\mathcal{Z}} \subseteq \mathcal{Z}$ which is a compact subset satisfying $\mu(\hat{\mathcal{Z}}) > 1 - \eta$. Suppose an algorithm is (K, ϵ) robust if restricted on $\hat{\mathcal{Z}}$, then on $\mathcal{Z}$ we have with probability at least $1 - \delta - \eta$,

$$|\mathcal{L}(\mathcal{A}_{\mathbf{s}}) - l_{\text{emp}}(\mathcal{A}_{\mathbf{s}})| \leq \epsilon(\mathbf{s}) + M\sqrt{\frac{2K \ln 2 + 2\ln(1/\delta)}{n}}.$$

Remark 3 Combining results presented in this section with Theorem 1 (or Theorem 4), it is easy to obtain generalization bounds for individual learning algorithms. It is of interest to understand the quality of robustness-based generalization bounds. For some learning algorithm, for example SVM and Lasso, the robustness bounds are indeed sub-optimal as it depends polynomially with respect to the covering number, while the bounds based on previous approaches typically depends polynomially with respect to the logarithm of the covering number. On the other hand, for some learning algorithms, the robustness based bound are indeed optimal. Example 3 is such a case, as it is known that its minimax excess error is of the order $\sqrt{K/n}$ (e.g., Devroye et al. 1996). Such noticeable difference can be attributed to the fact that in Lasso and SVM, there are redeeming features (small VC-dimensionality for Lasso and stability for SVM) that make them amenable to analysis. In general, when such features—ranging from sparseness to stability to having a small VC-dimesnionaility—do exist, a specialized analysis exploiting these features may yield tighter bounds than robustness-based analysis. When such features do not exist, a more generalized analysis based on robustness may turn to be the best choice.

7 Comparison and contrast with previous approaches

We devote this section to compare and contrast the proposed robustness-based approach with previously suggested approaches. Our goal is to demonstrate that the concept of robustness is essentially different from previous concepts, and it is possible to use the robust approach to derive stronger results than previous approaches in some cases. To this end, we present three results. First, we show that there are robust algorithms for classes with an infinite VC dimension. Similarly, we then show that there are robust, but not uniformly stable algorithms. Finally, we consider larger margin classification and show that robustness leads to a novel bound that implies a faster convergence rate than standard, fat-shattering dimension based results.

7.1 Robustness and VC dimension

We first investigate the relationship between robustness and VC-dimension. Indeed, it is easy to construct a robust algorithm whose *solution set* $\mathcal{H}^o = \{h \in \mathcal{H} | \exists \mathbf{s} : h = \mathcal{A}_{\mathbf{s}}\}$ has infinite VC dimensions. For example, consider the class of neural networks discussed in Example 7 (more precisely, the classification case as in Remark 1), which is shown to be robust. On the other hand, it is well-known that the set of neural networks (without a bound on the number of computational units) has infinite VC dimensions (Bartlett 1998), and hence is not amenable to VC-dimension based analysis.

7.2 Robustness and stability

We now consider the relationship between robustness and stability. While these notions are similar on an intuitive level, they are inherently different. To highlight this, we show by

example that a robust algorithm can be non-stable. This result is adapted from our previous work (Xu et al. 2010b), which among other things, showed that Lasso is not-stable. For completeness, we reproduce the results here. Recall the definition of uniform stability from Bousquet and Elisseeff (2002):

Definition 7 An algorithm $\mathbb{L}$ has a uniform stability bound of β_n with respect to the loss function l if the following holds:

$$\forall S \in \mathcal{Z}^n, \forall i \in \{1, \ldots, n\}, \quad \|l(\mathbb{L}_S, \cdot) - l(\mathbb{L}_{S^{\setminus i}}, \cdot)\|_\infty \leq \beta_n.$$

Here $\mathbb{L}_{S^{\setminus i}}$ stands for the learned solution with the ith sample removed from S.

An algorithm is stable when β_n decreases fast enough. Interestingly, Lasso, which is known to be robust, is non-stable: its uniform stability bound does not decrease at all, as the following theorem adapted from Xu et al. (2010b) shows.

Theorem 7 *Let $\mathcal{Z} = \mathcal{Y} \times \mathcal{X}$ be the sample space with m features, where $\mathcal{Y} \subseteq \mathbb{R}$, $\mathcal{X} \subseteq \mathbb{R}^m$, $0 \in \mathcal{Y}$ and $\mathbf{0} \in \mathcal{X}$. Let $\hat{\mathcal{Z}} = \mathcal{Y} \times \mathcal{X} \times \mathcal{X}$ be the sample space with $2m$ features. Then the uniform stability bound of Lasso is lower bounded by $\mathfrak{b}_n(\text{Lasso}, \mathcal{Z})$. Here, $\mathfrak{b}_n(\cdot, \cdot)$, termed* trivial bound, *is defined as*

$$\mathfrak{b}_n(\mathbb{L}, \mathcal{Z}) \triangleq \max_{(\mathbf{b}, A) \in \mathcal{Z}^n, \mathbf{z} \in \mathcal{X}} l(\mathbb{L}_{(\mathbf{b}, A)}, (0, \mathbf{z})).$$

Observe that $\mathfrak{b}_n(\mathbb{L}, \mathcal{Z}) \geq \mathfrak{b}_1(\mathbb{L}, \mathcal{Z})$ since by repeatedly choosing the worst sample (for $\mathfrak{b}_1$), the algorithm will yield the same solution. Hence the trivial bound does not diminish as the number of samples, n, increases. Thus, the uniform stability bound of Lasso does not decrease, which implies that Lasso is robust but non-stable.

7.3 Improved bounds through robustness

Finally, we show that robustness can lead to tighter generalization bounds. In particular, we consider generalization bounds of classification algorithms based on the margin achieved, and show that the robustness based bound is tighter than standard results based on fat-shattering dimension (Bartlett 1998). A fat-shattering dimension argument leads to the following result, adapted from Corollary 14 of Bartlett (1998):

Corollary 4 *Let $\mathcal{Y} = \{-1, +1\}$. Consider an arbitrary algorithm $\mathcal{A}$. With probability at least $1 - \delta$ over $\mathbf{s} \in \mathcal{Z}^n$, the following holds*

$$\mathcal{L}(\mathcal{A}_\mathbf{s}) \leq \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\left(\mathcal{D}(\mathbf{s}_i^{(x)}, \mathcal{A}_s) \leq \gamma, \text{ or } \mathcal{A}_\mathbf{s}(\mathbf{s}_i^{(x)}) \neq \mathbf{s}_i^{(y)}\right) + \sqrt{\frac{2}{n}(d \ln(34en/d) \log_2(578n) + \ln(4/\delta))},$$

where $d = \mathcal{N}(\gamma/16, \mathcal{X}, \rho)$.

On the other hand, combining Example 9 and Theorem 4 we have the following robustness-based bound.

Corollary 5 *Let $\mathcal{Y} = \{-1, +1\}$. Consider an arbitrary algorithm $\mathcal{A}$. With probability at least $1 - \delta$ over $\mathbf{s} \in \mathcal{Z}^n$, the following holds*

$$\mathcal{L}(\mathcal{A}_\mathbf{s}) \leq \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\big(\mathcal{D}(\mathbf{s}_i^{(x)}, \mathcal{A}_s) \leq \gamma, \text{ or } \mathcal{A}_\mathbf{s}(\mathbf{s}_i^{(x)}) \neq \mathbf{s}_i^{(y)}\big) + \sqrt{\frac{2}{n}(K \ln 2 + \ln(1/\delta))},$$

where $K = 2\mathcal{N}(\gamma/2, \mathcal{X}, \rho)$.

Since the first terms to both bounds are the same, we only need to compare the second terms. Neglecting constants, we have that

$$\sqrt{\frac{2}{n}(K \ln 2 + \ln(1/\delta))} \leq \epsilon; \quad \text{if } n \sim K/\epsilon^2.$$

As a comparison,

$$\sqrt{\frac{2}{n}(d \ln(34en/d) \log_2(578n) + \ln(4/\delta))} \leq \epsilon; \quad \text{if } n \sim d \ln^2(1/\epsilon)/\epsilon^2.$$

Note that $K \leq 2d$, hence we conclude that for the large margin classification algorithm, the robustness-based bound is tighter by a logarithmic factor than the fat-shattering based bound.

Remark 4 We remark that for specific margin-based algorithms, including boosting and neural networks, the bound in Corollary 4 has been improved using Radamacher complexity argument (Koltchinskii and Panchenko 2002; Kakade et al. 2009) so that the $\log n$ factor was removed. However, we are not aware of a similar improvement of margin-based bounds for general classification algorithms.

One advantage of the robustness approach is the geometric intuition it conveys. This often leads to significantly simplified proofs. For example, for large margin classification, we derived a bound that is (slightly) better than state-of-art results, with only a half page of argument. Moreover, and arguably more importantly, the geometric intuition makes it much easier to extend generalization results to interesting, non-standard learning setups such as Markovian samples or quantile loss.

8 Necessity of robustness

Thus far we have considered finite sample generalization bounds of robust algorithms. We now turn to asymptotic analysis. Our setup is as follows. We are given an increasing set of training samples $\mathbf{s} = (s_1, s_2, \ldots)$ and tested on an increasing set of testing samples $\mathbf{t} = (t_1, t_2, \ldots)$, where all samples are generated IID according to an unknown distribution μ. We use $\mathbf{s}(n)$ and $\mathbf{t}(n)$ to denote the first n elements of training samples and testing samples respectively. Therefore, $\mathbf{s}(n)$ and $\mathbf{t}(n)$ are random variables follow distribution μ^n, while $\mathbf{s}$ and $\mathbf{t}$ follow distribution μ^∞. In certain cases, we may fix a sequence of training samples, which we will denote by $\mathbf{s}^*$. Throughout this section, unless otherwise specified, $\Pr(\cdot)$ denotes the probability with respect to $\mathbf{t}$.

A learning method $\mathcal{A}$ is defined as a sequence of mappings $\{\mathcal{A}^k\}_{k\in\mathbb{N}}$ where $\mathcal{A}^k$ is a learning algorithm for a training sample set of size k, i.e.,

$$\mathcal{A}^k : \mathcal{Z}^k \mapsto \mathcal{H}.$$

For succinctness, we suppress the superscript whenever the number of training samples is clear. We use $L(\cdot,\cdot)$ to denote the average loss given a set of samples, i.e., for $h \in \mathcal{H}$,

$$L(h,\, \mathbf{t}(n)) \equiv \frac{1}{n}\sum_{i=1}^{n} l(h, t_i).$$

Recall that $\mathcal{L}(\cdot)$ denotes the expected loss, i.e.,

$$\mathcal{L}(h) = \mathbb{E}_{z\sim\mu} l(h, z).$$

We show in this section that robustness is an essential property of successful learning. In particular, a (weaker) notion of robustness characterizes generalizability, i.e., a learning algorithm generalizes if and only if it is weakly robust. To make this precise, we define the notion of generalizability and weak robustness first.

Definition 8 1. Given a sequence of training samples $\mathbf{s}^*$, a learning method $\mathcal{A}$ *generalizes w.r.t.* $\mathbf{s}^*$ if

$$\lim_n \left|\mathcal{L}(\mathcal{A}_{\mathbf{s}^*(n)}) - L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{s}^*(n))\right| = 0.$$

2. A learning method $\mathcal{A}$ *generalize w.p. 1* if it generalize w.r.t. almost all $\mathbf{s}$, where $\mathbf{s}$ contains IID samples following distribution μ.

We remark that the proposed notion of generalizability differs slightly from the standard one in the sense that the latter requires that the empirical risk and the expected risk converge in mean, while the proposed notion requires convergence w.p. 1. It is straightforward that the proposed notion implies the standard one.

Definition 9 1. Given a sequence of training samples $\mathbf{s}^*$, a learning method $\mathcal{A}$ is *weakly robust w.r.t.* $\mathbf{s}^*$ if there exists a sequence of $\{\mathcal{D}_n \subseteq \mathcal{Z}^n\}$ such that $\Pr(\mathbf{t}(n) \in \mathcal{D}_n) \to 1$, and

$$\lim_n \left\{ \max_{\hat{\mathbf{s}}(n)\in\mathcal{D}_n} \left| L(\mathcal{A}_{\mathbf{s}^*(n)}, \hat{\mathbf{s}}(n)) - L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{s}^*(n)) \right| \right\} = 0. \tag{6}$$

2. A learning method $\mathcal{A}$ is *a.s. weakly robust* if it is robust w.r.t. almost all $\mathbf{s}$.

In (6), $\hat{\mathbf{s}}(n)$ is any n-sample set belonging to $\mathcal{D}_n$, which intuitively can be regarded as a perturbed copy of the training sample set $\mathbf{s}^*(n)$. We briefly comment on the definition of weak robustness. Recall that the definition of robustness requires that the sample space can be partitioned into disjoint subsets such that if a testing sample belongs to the same partitioning set of a training sample, then they have similar loss. Weak robustness generalizes such notion by considering the average loss of testing samples and training samples. That is, if for a large (in the probabilistic sense) subset of $\mathcal{Z}^n$, the testing error is close to the training error, then the algorithm is weakly robust. It is easy to see, by Breteganolle-Huber-Carol lemma, that if for any fixed $\epsilon > 0$ there exists K such that $\mathcal{A}$ is (K, ϵ) robust, then $\mathcal{A}$ is weakly robust.

We now establish the main result of this section: weak robustness and generalizability are equivalent.

Theorem 8 *Fix a sequence of training samples* $\mathbf{s}^*$. *A learning method* $\mathcal{A}$ *generalizes w.r.t.* $\mathbf{s}^*$ *if and only if it is weakly robust w.r.t.* $\mathbf{s}^*$.

Proof We prove the sufficiency of weak robustness first. When $\mathcal{A}$ is weakly robust w.r.t. $\mathbf{s}^*$, by definition there exists $\{D_n\}$ such that for any δ, $\epsilon > 0$, there exists $N(\delta, \epsilon)$ such that for all $n > N(\delta, \epsilon)$, $\Pr(\mathbf{t}(n) \in D_n) > 1 - \delta$, and

$$\sup_{\hat{\mathbf{s}}(n) \in D_n} \left| L(\mathcal{A}_{\mathbf{s}^*(n)}, \hat{\mathbf{s}}(n)) - L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{s}^*(n)) \right| < \epsilon. \tag{7}$$

Therefore, the following holds for any $n > N(\delta, \epsilon)$,

$$\begin{aligned}
&\left| \mathcal{L}(\mathcal{A}_{\mathbf{s}^*(n)}) - L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{s}^*(n)) \right| \\
&= \left| \mathbb{E}_{\mathbf{t}(n)}(L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{t}(n))) - L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{s}^*(n)) \right| \\
&= \Big| \Pr(\mathbf{t}(n) \notin D_n) \mathbb{E}(L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{t}(n)) | \mathbf{t}(n) \notin D_n) \\
&\quad + \Pr(\mathbf{t}(n) \in D_n) \mathbb{E}(L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{t}(n)) | \mathbf{t}(n) \in D_n) - L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{s}^*(n)) \Big| \\
&\leq \Pr(\mathbf{t}(n) \notin D_n) \Big| \mathbb{E}(L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{t}(n)) \Big| \mathbf{t}(n) \notin D_n) - L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{s}^*(n)) \Big| \\
&\quad + \Pr(\mathbf{t}(n) \in D_n) \Big| \mathbb{E}(L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{t}(n)) | \mathbf{t}(n) \in D_n) - L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{s}^*(n)) \Big| \\
&\leq \delta M + \sup_{\hat{\mathbf{s}}(n) \in D_n} \left| L(\mathcal{A}_{\mathbf{s}^*(n)}, \hat{\mathbf{s}}(n)) - L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{s}^*(n)) \right| \leq \delta M + \epsilon.
\end{aligned}$$

Here, the first equality holds because the testing samples $\mathbf{t}(n)$ consists of n i.i.d. samples following μ. The second equality holds by conditional expectation. The last inequalities hold due to the assumption that the loss function is upper bounded by M, as well as (7).

We thus conclude that the algorithm $\mathcal{A}$ generalizes for $\mathbf{s}^*$, because ϵ and δ can be arbitrary.

Now we turn to the necessity of weak robustness. First, we establish the following lemma.

Lemma 2 *Given* $\mathbf{s}^*$, *if a learning method* $\mathcal{A}$ *is not weakly robust w.r.t.* $\mathbf{s}^*$, *then there exists* ϵ^*, $\delta^* > 0$ *such that the following holds for infinitely many* n,

$$\Pr\left(\left| L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{t}(n)) - L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{s}^*(n)) \right| \geq \epsilon^* \right) \geq \delta^*. \tag{8}$$

Proof We prove the lemma by contradiction. Assume that such ϵ^* and δ^* do not exist. Let $\epsilon_v = \delta_v = 1/v$ for $v = 1, 2, \ldots$, then there exists a non-decreasing sequence $\{N(v)\}_{v=1}^{\infty}$ such that for all v, if $n \geq N(v)$ then $\Pr(|L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{t}(n)) - L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{s}^*(n))| \geq \epsilon_v) < \delta_v$. For each n, define the following set:

$$\mathcal{D}_n^v \triangleq \{\hat{\mathbf{s}}(n) \,|\, |L(\mathcal{A}_{\mathbf{s}^*(n)}, \hat{\mathbf{s}}(n)) - L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{s}^*(n))| < \epsilon_v\}.$$

Thus, for $n \geq N(v)$ we have

$$\Pr(\mathbf{t}(n) \in \mathcal{D}_n^v) = 1 - \Pr\left(\left| L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{t}(n)) - L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{s}^*(n)) \right| \geq \epsilon_v \right) > 1 - \delta_v.$$

For $n \geq N(1)$, define $\mathcal{D}_n \triangleq \mathcal{D}_n^{v(n)}$, where: $v(n) \triangleq \max(v | N(t) \leq n;\ v \leq n)$. Thus for all $n \geq N(1)$ we have that $\Pr(\mathbf{t}(n) \in \mathcal{D}_n) > 1 - \delta_{v(n)}$ and $\sup_{\hat{\mathbf{s}}(n) \in \mathcal{D}_n} |L(\mathcal{A}_{\mathbf{s}^*(n)}, \hat{\mathbf{s}}(n)) -$

$L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{s}^*(n))| < \epsilon_{v(n)}$. Note that $v(n) \uparrow \infty$, it follows that $\delta_{v(n)} \to 0$ and $\epsilon_{v(n)} \to 0$. Therefore, $\Pr(\mathbf{t}(n) \in \mathcal{D}_n) \to 1$, and

$$\lim_{n\to\infty} \left\{ \sup_{\hat{\mathbf{s}}(n)\in\mathcal{D}_n} \left| L(\mathcal{A}_{\mathbf{s}^*(n)}, \hat{\mathbf{s}}(n)) - L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{s}^*(n)) \right| \right\} = 0.$$

That is, $\mathcal{A}$ is weakly robust w.r.t. $\mathbf{s}$, which is a desired contradiction. □

We now prove the necessity of weak robustness. Recall that $l(\cdot,\cdot)$ is uniformly bounded. Thus by Hoeffding's inequality we have that for any ϵ, δ, there exists n^* such that for any $n > n^*$, with probability at least $1-\delta$, we have $|\frac{1}{n}\sum_{i=1}^{n} l(\mathcal{A}_{\mathbf{s}^*(n)}, t_i) - \mathcal{L}(l(\mathcal{A}_{\mathbf{s}^*(n)})| \le \epsilon$. This implies that

$$L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{t}(n)) - \mathcal{L}(\mathcal{A}_{\mathbf{s}^*(n)}) \xrightarrow{\Pr} 0. \tag{9}$$

Since algorithm $\mathcal{A}$ is not robust, Lemma 2 implies that (8) holds for infinitely many n. This, combined with (9) implies that for infinitely many n,

$$\left| \mathcal{L}(\mathcal{A}_{\mathbf{s}^*(n)}, t) - L(\mathcal{A}_{\mathbf{s}^*(n)}, \mathbf{s}^*(n)) \right| \ge \frac{\epsilon^*}{2},$$

which means that $\mathcal{A}$ does not generalize. Thus, the necessity of weak robustness is established. □

Theorem 8 immediately leads to the following corollary.

Corollary 6 *A learning method $\mathcal{A}$ generalizes w.p. 1 if and only if it is a.s. weakly robust.*

Remark 5 In Shalev-Shwartz et al. (2009), the authors investigated a closely related problem, namely, "when is a problem *learnable*?" More precisely, learnability is defined as follows.

Definition 10 (Adapted from Shalev-Shwartz et al. 2009) A learning problem defined through a set of hypothesis $\mathcal{H}$ and a loss function $l(\cdot,\cdot)$ is *learnable*, if there exists a learning method $\mathcal{A}$ and a monotone decreasing sequence $\epsilon_{\text{cons}}(n) \downarrow 0$ such that

$$\forall \mu: \quad \mathbb{E}_{\mathbf{s}(n)\sim\mu^n}[\mathbb{E}_{z\sim\mu} l(\mathcal{A}_{\mathbf{s}(n)}, z) - \inf_{h\in\mathcal{H}} \mathbb{E}_{z\sim\mu} l(h, Z)] \le \epsilon_{\text{cons}}(n).$$

The authors then showed that learnability can be characterized by a version of stability, in the general learning setup. It is worthwhile to note the difference between learnability and generalizability that is investigated in this section, namely, learnability requires the excess risk to converge *uniformly w.r.t. all distributions*, while generalizability requires the generalization gap to converge, but the rate can vary for different distributions. As such, learnability is a more strict condition. Indeed, as Shalev-Shwartz et al. (2009) noted, for the supervised learning case (arguably the most common learning setup), classical results stated that learnability is equivalent to finiteness of the VC dimension of $\mathcal{H}$ (Vapnik and Chervonenkis 1974). In contrast, the characterization of the generalizability seems to be an open question, even in the supervised learning setup. Thus, to the best of our knowledge, *weak robustness* provides the very first attempt to answer it.

Zakai and Ritov (2009) proposed a notion termed *localizability* and showed that a supervised-learning algorithm is *uniformly consistent* if and only if it simultaneously satisfies two conditions: first, the algorithm correctly estimates the true mean asymptotically, and second, it satisfies a form of localizability. Roughly speaking, an algorithm is localizable if the prediction of a testing sample does not change significantly when the algorithm is trained on those training samples that are close to the testing sample. Beyond some apparent difference in the setup (e.g., consistency vs generalizability, supervised learning vs general learning, etc.), the main difference between localizability and robustness is that localizability requires that the outputs of two runs of an algorithm—one on the entire training set and the other on a subset of training samples—are "close" to each other. In this spirit, localizability is a notion close to stability. In contrast, robustness considers performance for different testing runs for *one* output solution of an algorithm. Despite these differences though, in a high-level, it appears that both localizability and robustness are *geometric* notations that are critical to the performance of learning algorithms. Therefore, it would be interesting to investigate the relationship between these two notions. Due to space constraints, a detailed investigation is out of the scope of this paper.

9 Conclusions and future directions

In this paper we investigated the generalization ability of learning algorithm based on their robustness: the property that if a testing sample is "similar" to a training sample, then its loss is close to the training error. This provides a novel approach, different from the complexity or stability argument, in studying the performance of learning algorithms. We further showed that a weak notion of robustness characterizes generalizability, which implies that robustness is a fundamental property for learning algorithms to work.

Before concluding the paper, we outline several directions for future research.

- *Adaptive partition:* In Definition 2 when the notion of robustness was introduced, we required that the partitioning of $\mathcal{Z}$ into K sets is *fixed*. That is, regardless of the training sample set, we partition $\mathcal{Z}$ into the same K sets. A natural and interesting question is what if such fixed partition does not exist, while instead we can only partition $\mathcal{Z}$ into K sets *adaptively*, i.e., for different training set we will have a different partitioning of $\mathcal{Z}$. Adaptive partition can be used to study algorithms such as k-NN. Our current proof technique does not straightforwardly extend to such a setup, and we would like to understand whether a meaningful generalization bound under this weaker notion of robustness can be obtained. We note that for the standard learning setup, where samples are IID, robustness-based argument seems to often lead to generalization bounds not superior than previous approaches. One important future research direction would be to examine the possibility of using adaptive partition to obtain tighter bounds.
- *Mismatched datasets:* One advantage of algorithmic robustness framework is the ability to handle non-standard learning setups. For example, in Sect. 4.1 we derived generalization bounds for quantile loss. A problem of the same essence is the *mismatched datasets*, also known as *domain adaption* (e.g., Ben-David et al. 2007; Mansour et al. 2009 and reference therein). Here the training samples are generated according to a distribution slightly different from that of the testing samples, e.g., the two distributions may have a small K-L divergence. Indeed, following a similar argument as the proof of Theorem 1, one can show that if algorithm $\mathcal{A}$ is (K, ϵ)-robust w.r.t. partition $C_1, \ldots, C_K$, the training

samples are iid following μ_s, we can bound the error w.r.t. a distribution μ_t by

$$\mathbb{E}_{z\sim\mu_t} l(\mathcal{A}_{\mathbf{s}}, z) \leq l_{\text{emp}}(\mathcal{A}_{\mathbf{s}}) + \epsilon + M\sqrt{\frac{2K\ln 2 + 2\ln(1/\delta)}{n}} + \sum_{i=1}^{K} |\mu_t(C_i) - \mu_s(C_i)|.$$

Note that the last term, which bounds the error due to domain adaption, only depends on the difference of μ_t and μ_s over $\{C_i\}$. This is upper bounded by, and can be much smaller than, the total variation of μ_s and μ_t defined as

$$d_{L_1}(\mu_s, \mu_t) = 2\sup_{B\in\mathcal{B}} |\mu_s(B) - \mu_t(B)|,$$

where $\mathcal{B}$ is the set of all measurable sets in $\mathcal{Z}$. Since the total variation is commonly used in domain adaption, the algorithmic robustness approach may lead to better generalization bounds for domain adaption.

– *Outlier removal:* One possible reason that the training samples is generated differently from the testing sample is outlier corruption. It is often the case that the training sample set is corrupted by some outliers. In addition, algorithms designed to be outlier resistent abound in the literature (e.g., Huber 1981; Rousseeuw and Leroy 1987). The robustness framework may provide a novel approach in studying both the generalization ability and the outlier resistent property of these algorithms. In particular, the results reported in Sect. 4.1 can serve as a starting point of future research in this direction.
– *Other robust algorithms:* The proposed robust approach considers a general learning setup. However, except for PCA, the algorithms investigated in Sect. 6 all belong to the supervised learning setting. One natural extension is to investigate other robust unsupervised and semi-supervised learning algorithms. One difficulty is that compared to supervised learning case, the analysis of unsupervised/semi-supervised learning algorithms can be challenging, due to the fact that many of them are random iterative algorithms (e.g., k-means).

Acknowledgements We thank the reviewers for detailed comments that significantly improves the exposition of the paper. The research of H. Xu is partially supported by the NUS startup grant (R-265-000-384-133). The research of S. Mannor is partially supported by the Israel Science Foundation (contract 890015).

Appendix A: Proof of Theorem 3

For succinctness, let

$$\lambda_0 \triangleq \frac{2T}{\alpha n} + \sqrt{\frac{2T^2(K\ln 2 + \ln(1/\delta))}{\alpha^2 n}}.$$

Observe that $\lambda_0 > 2T/(\alpha n)$, which leads to

$$n > \frac{2T}{\alpha\lambda_0}.$$

Let N_i be the set of index of points of $\mathbf{s}$ that fall into the C_i. Consider the set of functions $\mathcal{H} = \{\mathbf{1}(\mathbf{x} \in H) | H = \bigcup_{i\in I} C_i;\ \forall I \subseteq \{1,\ldots,K\}\}$, i.e., the set of indicator functions of all different unions of C_i. Then $|\mathcal{H}| = 2^K$. Furthermore, fix a $h_0 \in \mathcal{H}$,

$$\Pr\left(\sum_{j=1}^{K}\left|\frac{|N_j|}{n}-\pi(C_j)\right|\geq\lambda\right)=\Pr\left\{\sup_{h\in\mathcal{H}}\left[\frac{1}{n}\sum_{i=1}^{n}h(s_i)-\mathbb{E}_\pi h(s)\right]\geq\lambda\right\}$$
$$\leq 2^K\Pr\left[\frac{1}{n}\sum_{i=1}^{n}h_0(s_i)-\mathbb{E}_\pi h_0(s)\geq\lambda\right].$$

Since $\|h_0\|_\infty=1$ and recall $n>2T/\lambda\alpha$, we can apply Lemma 1 to get

$$\Pr\left[\frac{1}{n}\sum_{i=1}^{n}h_0(s_i)-\mathbb{E}_\pi h_0(s)\geq\lambda\right]\leq\exp\left(-\frac{\alpha^2(n\lambda-2T/\alpha)^2}{2nT^2}\right).$$

Substitute in λ_0,

$$\Pr\left(\sum_{j=1}^{K}\left|\frac{|N_j|}{n}-\pi(C_j)\right|\geq\lambda_0\right)\leq 2^K\exp\left(-\frac{\alpha^2(n\lambda_0-2T/\alpha)^2}{2nT^2}\right)=\delta.$$

Thus, following an identical argument as the proof of Theorem 1, we have with probability $1-\delta$,

$$|\mathcal{L}(\mathcal{A}_\mathbf{s})-l_{\text{emp}}(\mathcal{A}_\mathbf{s})|\leq\epsilon(\mathbf{s})+M\lambda_0$$
$$=\epsilon(\mathbf{s})+M\left\{\frac{2T}{\alpha n}+\sqrt{\frac{2T^2(K\ln 2+\ln(1/\delta))}{\alpha^2 n}}\right\}.$$

Appendix B: Proof of Theorem 4

Let N_i and $\hat{N}_i$ be the set of indices of points of $\mathbf{s}$ and $\hat{\mathbf{s}}$ that fall into the C_i, respectively. Similarly to the proof of Theorem 1, we note that $(|N_1|,\ldots,|N_K|)$ is a multinomial random variable with parameters n and $(\mu(C_1),\ldots,\mu(C_K))$. And hence due to Breteganolle-Huber-Carol inequality, the following holds with probability at least $1-\delta$,

$$\sum_{i=1}^{K}\left|\frac{|N_i|}{n}-\mu(C_i)\right|\leq\sqrt{\frac{2K\ln 2+2\ln(1/\delta)}{n}}. \tag{10}$$

Furthermore, we have

$$|\mathcal{L}(\mathcal{A}_\mathbf{s})-l_{\text{emp}}(\mathcal{A}_\mathbf{s})|$$
$$=\left|\sum_{i=1}^{K}\mathbb{E}(l(\mathcal{A}_\mathbf{s},z)|z\in C_i)\mu(C_i)-\frac{1}{n}\sum_{i=1}^{n}l(\mathcal{A}_\mathbf{s},s_i)\right|$$
$$\leq\left|\sum_{i=1}^{K}\mathbb{E}(l(\mathcal{A}_\mathbf{s},z)|z\in C_i)\frac{|N_i|}{n}-\frac{1}{n}\sum_{i=1}^{n}l(\mathcal{A}_\mathbf{s},s_i)\right|$$
$$+\left|\sum_{i=1}^{K}\mathbb{E}(l(\mathcal{A}_\mathbf{s},z)|z\in C_i)\mu(C_i)-\sum_{i=1}^{K}\mathbb{E}(l(\mathcal{A}_\mathbf{s},z)|z\in C_i)\frac{|N_i|}{n}\right|$$

$$\leq \left| \frac{1}{n} \sum_{i=1}^{K} \left[|N_i| \times \mathbb{E}(l(\mathcal{A}_{\mathbf{s}}, z) | z \in C_i) - \sum_{j \in \hat{N}_i} l(\mathcal{A}_{\mathbf{s}}, s_j) - \sum_{j \in N_i, j \notin \hat{N}_i} l(\mathcal{A}_{\mathbf{s}}, s_j) \right] \right|$$

$$+ \left| \max_{z \in \mathcal{Z}} |l(\mathcal{A}_{\mathbf{s},z})| \sum_{i=1}^{K} \left| \frac{|N_i|}{n} - \mu(C_i) \right| \right|.$$

Note that due to the triangle inequality as well as the assumption that the loss is non-negative and upper bounded by M, the right-hand side can be upper bounded by

$$\left| \frac{1}{n} \sum_{i=1}^{K} \sum_{j \in \hat{N}_i} \max_{z_2 \in C_i} |l(\mathcal{A}_{\mathbf{s}}, s_j) - l(\mathcal{A}_{\mathbf{s}}, z_2)| \right| + \left| \frac{1}{n} \sum_{i=1}^{K} \sum_{j \in N_i, j \notin \hat{N}_i} \max_{z_2 \in C_i} |l(\mathcal{A}_{\mathbf{s}}, s_j) - l(\mathcal{A}_{\mathbf{s}}, z_2)| \right|$$

$$+ M \sum_{i=1}^{K} \left| \frac{|N_i|}{n} - \mu(C_i) \right|$$

$$\leq \frac{\hat{n}(\mathbf{s})}{n} \epsilon(\mathbf{s}) + \frac{n - \hat{n}(\mathbf{s})}{n} M + M \sum_{i=1}^{K} \left| \frac{|N_i|}{n} - \mu(C_i) \right|,$$

where the inequality holds due to definition of N_i and $\hat{N}_i$. The theorem follows by applying (10).

Appendix C: Proof of Theorem 5

We observe the following properties of quantile value and truncated mean:

1. If X is supported on $\mathbb{R}^+$ and $\beta_1 \geq \beta_2$, then

$$\mathbb{Q}^{\beta_1}(X) \geq \mathbb{Q}^{\beta_2}(X); \qquad \mathbb{T}^{\beta_1}(X) \geq \mathbb{T}^{\beta_2}(X).$$

2. If Y stochastically dominates X, i.e., $\Pr(Y \geq a) \geq \Pr(X \geq a)$ for all $a \in \mathbb{R}$, then for any β,

$$\mathbb{Q}^{\beta}(Y) \geq \mathbb{Q}^{\beta}(X); \qquad \mathbb{T}^{\beta}(Y) \geq \mathbb{T}^{\beta}(X).$$

3. The β-truncated mean of empirical distribution of nonnegative $(x_1, \ldots, x_n)$ is given by

$$\min_{\alpha: 0 \leq \alpha_i \leq 1/n, \sum_{i=1}^{n} \alpha_i \leq \beta} \sum_{i=1}^{n} \alpha_i x_i.$$

By definition of pseudo-robustness, $\mathcal{Z}$ can be partitioned into K disjoint sets, denoted as $\{C_i\}_{i=1}^{K}$, and a subset of training samples $\hat{\mathbf{s}}$ with $|\hat{\mathbf{s}}| = \hat{n}(\mathbf{s})$ such that

$$z_1 \in \hat{\mathbf{s}},\ z_1, z_2 \in C_i, \quad \Longrightarrow \quad |l(\mathcal{A}_{\mathbf{s}}, z_1) - l(\mathcal{A}_{\mathbf{s}}, z_2)| \leq \epsilon(\mathbf{s}); \quad \forall s.$$

Let N_i be the set of index of points of $\mathbf{s}$ that fall into the C_i. Let $\mathcal{E}$ be the event that the following holds:

$$\sum_{i=1}^{K} \left| \frac{|N_i|}{n} - \mu(C_i) \right| \leq \sqrt{\frac{2K \ln 2 + 2 \ln(1/\delta)}{n}}.$$

From the proof of Theorem 1, $\Pr(\mathcal{E}) \geq 1-\delta$. Hereafter we restrict the discussion to the case when $\mathcal{E}$ holds.

Denote

$$v_j = \arg\min_{z \in \mathcal{C}_j} l(\mathcal{A}_{\mathbf{s}}, z).$$

By symmetry, without loss of generality we assume that $0 \leq l(\mathcal{A}_{\mathbf{s}}, v_1) \leq l(\mathcal{A}_{\mathbf{s}}, v_2) \leq \cdots \leq l(\mathcal{A}_{\mathbf{s}}, v_K) \leq M$. Define a set of samples $\tilde{\mathbf{s}}$ as

$$\tilde{s}_i = \begin{cases} s_i & \text{if } s_i \in \hat{\mathbf{s}}; \\ v_j & \text{if } s_i \notin \hat{\mathbf{s}},\ s_i \in \mathcal{C}_j. \end{cases}$$

Define discrete probability measures $\hat{\mu}$ and $\tilde{\mu}$, supported on $\{v_1, \ldots, v_K\}$ as

$$\hat{\mu}(\{v_j\}) = \mu(\mathcal{C}_j); \quad \tilde{\mu}(\{v_j\}) = \frac{|N_j|}{n}.$$

Further, let $\tilde{\mu}_{\text{emp}}$ denote the empirical distribution of sample set $\tilde{\mathbf{s}}$.

Proof of (I):

Observe that μ stochastically dominates $\hat{\mu}$, hence

$$\mathcal{Q}(\mathcal{A}_{\mathbf{s}}, \beta, \hat{\mu}) \leq \mathcal{Q}(\mathcal{A}_{\mathbf{s}}, \beta, \mu). \tag{11}$$

Also by definition of $\mathcal{Q}(\cdot)$ and $\hat{\mu}$,

$$\mathcal{Q}(\mathcal{A}_{\mathbf{s}}, \beta, \hat{\mu}) = v_{k^*}; \quad \text{where: } k^* = \min\left\{k : \sum_{i=1}^{k} \hat{\mu}(v_i) \geq \beta\right\}.$$

Let $\overline{\mathbf{s}}$ be the set of all samples s_i such that $s_i \in \hat{\mathbf{s}}$, and $s_i \in \mathcal{C}_j$ for some $j \leq k^*$. Observe that

$$\forall s_i \in \hat{\mathbf{s}} : l(\mathcal{A}_{\mathbf{s}}, s_i) \leq v_{k^*} + \epsilon(\mathbf{s}) = \mathcal{Q}(\mathcal{A}_{\mathbf{s}}, \beta, \hat{\mu}) + \epsilon(\mathbf{s}). \tag{12}$$

Note that $\mathcal{E}$ implies

$$\frac{1}{n}\sum_{j=1}^{k^*}\sum_{s_i \in \mathcal{C}_j} 1 \geq \sum_{j=1}^{k^*} \mu(\mathcal{C}_j) - \lambda_0 = \sum_{j=1}^{k} \hat{\mu}(v_j) - \lambda_0 \geq \beta - \lambda_0.$$

Since $\mathcal{A}_{\mathbf{s}}$ is pseudo robust, we have

$$\frac{1}{n}\sum_{s_i \notin \hat{\mathbf{s}}} = \frac{n - \hat{n}(\mathbf{s})}{n}.$$

Therefore

$$\frac{1}{n}\sum_{j=1}^{k^*}\sum_{s_i \in \overline{\mathbf{s}}, s_i \in \mathcal{C}_j} 1 \geq \frac{1}{n}\sum_{j=1}^{k^*}\sum_{s_i \in \mathcal{C}_j} 1 - \frac{1}{n}\sum_{s_i \notin \hat{\mathbf{s}}} 1 \geq \beta - \lambda_0 - \frac{n - \hat{n}(\mathbf{s})}{n}.$$

Thus, $\overline{\mathbf{s}}$ is a subset of $\mathbf{s}$ of at least $n(\beta - \lambda_0 - (n - \hat{n}(\mathbf{s}))/n)$ elements. Thus (11) and (12) lead to

$$\mathcal{Q}(\mathcal{A}_{\mathbf{s}}, \beta - \lambda_0 - (n - \hat{n}(\mathbf{s}))/n, \mu_{\text{emp}}) \leq \max\{s_i : s_i \in \overline{\mathbf{s}}\} \leq \mathcal{Q}(\mathcal{A}_{\mathbf{s}}, \beta, \mu) + \epsilon(\mathbf{s}).$$

Thus, we establish the left inequality. The proof of the right one is identical and hence omitted.

Proof of (II):

The proof constitutes four steps.

Step 1: Observe that μ stochastically dominates $\hat{\mu}$, hence

$$\mathcal{T}(\mathcal{A}_{\mathbf{s}}, \beta, \hat{\mu}) \leq \mathcal{T}(\mathcal{A}_{\mathbf{s}}, \beta, \mu).$$

Step 2: We prove that

$$\mathcal{T}(\mathcal{A}_{\mathbf{s}}, \beta - \lambda_0, \tilde{\mu}) \leq \mathcal{T}(\mathcal{A}_{\mathbf{s}}, \beta, \hat{\mu}).$$

Note that t $\mathcal{E}$ implies for all j, we have

$$\tilde{\mu}(\{v_1, \ldots, v_j\}) - \lambda_0 \leq \hat{\mu}(\{v_1, \ldots, v_j\}).$$

Therefore, there uniquely exists a non-negative integer j^* and a $c^* \in [0, 1)$ such that

$$\hat{\mu}(\{v_1, \ldots, v_{j^*}\}) + c^* \hat{\mu}(\{v_{j^*+1}\}) = \beta,$$

and define

$$\hat{\beta} = \sum_{i=1}^{j^*} \min(\tilde{\mu}(\{v_i\}), \hat{\mu}(\{v_i\})) + c^* \min(\tilde{\mu}(\{v_{j^*+1}\}), \hat{\mu}(\{v_{j^*+1}\})), \tag{13}$$

then we have $\hat{\beta} \geq \beta - \lambda_0$, which leads to

$$\begin{aligned}
&\mathcal{T}(\mathcal{A}_{\mathbf{s}}, \beta - \lambda_0, \tilde{\mu}) \leq \mathcal{T}(\mathcal{A}_{\mathbf{s}}, \hat{\beta}, \tilde{\mu}) \\
&\overset{(a)}{\leq} \sum_{i=1}^{j^*} l(\mathcal{A}_{\mathbf{s}}, v_i) \min(\tilde{\mu}(\{v_i\}), \hat{\mu}(\{v_i\})) + c^* l(\mathcal{A}_{\mathbf{s}}, v_{j^*+1}) \min(\tilde{\mu}(\{v_{j^*+1}\}), \hat{\mu}(\{v_{j^*+1}\})) \\
&\leq \sum_{i=1}^{j^*} l(\mathcal{A}_{\mathbf{s}}, v_i) \hat{\mu}(\{v_i\}) + c^* l(\mathcal{A}_{\mathbf{s}}, v_{j^*+1}) \hat{\mu}(\{v_{j^*+1}\}) = \mathcal{T}(\mathcal{A}_{\mathbf{s}}, \beta, \hat{\mu}),
\end{aligned}$$

where (a) holds because (13) essentially means that $\mathcal{T}(\mathcal{A}_{\mathbf{s}}, \hat{\beta}, \tilde{\mu})$ is a weighted sum with total weights equals to $\hat{\beta}$, which puts more weights on small terms, and hence is smaller.

Step 3: We prove that

$$\mathcal{T}(\mathcal{A}_{\mathbf{s}}, \beta - \lambda_0, \tilde{\mu}_{\text{emp}}) - \epsilon(\mathbf{s}) \leq \mathcal{T}(\mathcal{A}_{\mathbf{s}}, \beta - \lambda_0, \tilde{\mu}).$$

Let $\tilde{\mathbf{t}}$ be a set of n samples, such that N_j of them are v_j for $j = 1, \ldots, K$. Observe that $\tilde{\mu}$ is the empirical distribution of $\tilde{\mathbf{t}}$. Further note that there is a one-to-one mapping between samples in $\tilde{\mathbf{s}}$ and that in $\tilde{\mathbf{t}}$ such that each pair (say $\tilde{s}_i, \tilde{t}_i$) of samples belongs to the same $\mathcal{C}_j$. By definition of $\tilde{\mathbf{s}}$ this guarantees that $|l(\mathcal{A}_{\mathbf{s}}, \tilde{s}_i) - l(\mathcal{A}_{\mathbf{s}}, \tilde{t}_i)| \leq \epsilon(\mathbf{s})$, which implies

$$\mathcal{T}(\mathcal{A}_{\mathbf{s}}, \beta - \lambda_0, \tilde{\mu}_{\text{emp}}) - \epsilon(\mathbf{s}) \leq \mathcal{T}(\mathcal{A}_{\mathbf{s}}, \beta - \lambda_0, \tilde{\mu}).$$

Step 4: We prove that

$$\mathcal{T}\left(\mathcal{A}_{\mathbf{s}}, \beta - \lambda_0 - \frac{n - \hat{n}(\mathbf{s})}{n}, \mu_{\text{emp}}\right) \leq \mathcal{T}(\mathcal{A}_{\mathbf{s}}, \beta - \lambda_0, \tilde{\mu}_{\text{emp}}).$$

Let $\mathbb{I} = \{i : s_i = \tilde{s}_i\}$), the following holds:

$$\sum_{i=1}^{n} \alpha_i l(\mathcal{A}_{\mathbf{s}}, \tilde{s}_i) \geq \sum_{i \in \mathbb{I}} \alpha_i l(\mathcal{A}_{\mathbf{s}}, \tilde{s}_i) = \sum_{i \in \mathbb{I}} \alpha_i l(\mathcal{A}_{\mathbf{s}}, s_i); \quad \forall \alpha : 0 \leq \alpha_i \leq \frac{1}{n}; \sum_{i=1}^{n} \alpha_i = \beta - \lambda_0.$$

Note that $|\{i \notin \mathbb{I}\}| = n - \hat{n}(\mathbf{s})$, then $\sum_{i \in \mathbb{I}} \alpha_i \geq \beta - \lambda_0 - \frac{n - \hat{n}(\mathbf{s})}{n}$. Thus we have $\forall \alpha : 0 \leq \alpha_i \leq \frac{1}{n}$; $\sum_{i=1}^{n} \alpha_i = \beta - \lambda_0$,

$$\sum_{i \in \mathbb{I}} \alpha_i l(\mathcal{A}_{\mathbf{s}}, s_i) \geq \min_{\alpha' : 0 \leq \alpha'_i \leq \frac{1}{n}, \sum_{i=1}^{n} \alpha'_i \leq \beta - \lambda_0 - \frac{n - \hat{n}(\mathbf{s})}{n}} \sum_{i=1}^{n} \alpha'_i l(\mathcal{A}_{\mathbf{s}}, s_i) = \mathcal{T}(\mathcal{A}_{\mathbf{s}}, \beta - \lambda_0, \tilde{\mu}_{\text{emp}}).$$

Therefore,

$$\sum_{i=1}^{n} \alpha_i l(\mathcal{A}_{\mathbf{s}}, \tilde{s}_i) \geq \mathcal{T}\left(\mathcal{A}_{\mathbf{s}}, \beta - \lambda_0 - \frac{n - \hat{n}(\mathbf{s})}{n}, \mu_{\text{emp}}\right); \quad \forall \alpha : 0 \leq \alpha_i \leq \frac{1}{n}; \sum_{i=1}^{n} \alpha_i = \beta - \lambda_0.$$

Minimization over α on both side. We proved

$$\mathcal{T}\left(\mathcal{A}_{\mathbf{s}}, \beta - \lambda_0 - \frac{n - \hat{n}(\mathbf{s})}{n}, \mu_{\text{emp}}\right) \leq \mathcal{T}(\mathcal{A}_{\mathbf{s}}, \beta - \lambda_0, \tilde{\mu}_{\text{emp}}).$$

Combining all four steps, we proved the left inequality, i.e.,

$$\mathcal{T}\left(\mathcal{A}_{\mathbf{s}}, \beta - \lambda_0 - \frac{n - \hat{n}(\mathbf{s})}{n}, \mu_{\text{emp}}\right) - \epsilon(\mathbf{s}) \leq \mathcal{T}(\mathcal{A}_{\mathbf{s}}, \beta, \mu).$$

The right inequality can be proved identically and hence omitted.

Appendix D: Proof of Example 3

We can partition $\mathcal{Z}$ as $\{-1\} \times \mathcal{C}_1, \ldots, \{-1\} \times \mathcal{C}_K, \{+1\} \times \mathcal{C}_1, \ldots, \{+1\} \times \mathcal{C}_K$. Consider z_a, z_b that belong to a same set, then $z_a^{(y)} = z_b^{(y)}$, and $\exists i$ such that $z_a^{(x)}, z_b^{(x)} \in \mathcal{C}_i$, which by the definition of Majority Voting algorithm implies that $\mathcal{A}_{\mathbf{s}}(z_a^{(x)}) = \mathcal{A}_{\mathbf{s}}(z_b^{(x)})$. Thus, we have

$$l(\mathcal{A}_{\mathbf{s}}, z_a) = f(z_a^{(y)}, \mathcal{A}_{\mathbf{s}}(z_a^{(x)})) = f(z_b^{(y)}, \mathcal{A}_{\mathbf{s}}(z_b^{(x)})) = l(\mathcal{A}_{\mathbf{s}}, z_b).$$

Hence MV is $(2K, 0)$-robust.

Appendix E: Proof of Example 5

The existence of $f_{\mathbb{H}}(\gamma)$ follows from the compactness of $\mathcal{X}$ and continuity of $k(\cdot, \cdot)$.

To prove the robustness of SVM, let (w^*, d^*) be the solution given training data $\mathbf{s}$. To avoid notation clutter, let $y_i = s_i^{(y)}$ and $x_i = s_i^{(x)}$. Thus, we have (due to optimality of w^*, d^*)

$$c\|w^*\|_{\mathbb{H}}^2 + \frac{1}{n}\sum_{i=1}^{n}[1 - y_i(\langle w^*, \phi(x_i)\rangle + d^*)]^+ \leq c\|0\|_{\mathbb{H}}^2 + \frac{1}{n}\sum_{i=1}^{n}[1 - y_i(\langle 0, \phi(x_i)\rangle + 0)]^+ = 1,$$

which implies $\|w^*\|_{\mathbb{H}} \leq \sqrt{1/c}$. Let $c_1, \ldots, c_{\mathcal{N}(\gamma/2, \mathcal{X}, \|\cdot\|_2)}$ be a $\gamma/2$-cover of $\mathcal{X}$ (recall that $\mathcal{X}$ is compact), then we can partition $\mathcal{Z}$ as $2\mathcal{N}(\gamma/2, \mathcal{X}, \|\cdot\|_2)$ sets, such that if (y_1, x_1) and (y_2, x_2) belongs to the same set, then $y_1 = y_2$ and $\|x_1 - x_2\|_2 \leq \gamma/2$.

Further observe that if $y_1 = y_2$ and $\|x_1 - x_2\|_2 \leq \gamma/2$, then

$$\begin{aligned}
&|l((w^*, d^*), z_1) - l((w^*, d^*), z_2))| \\
&\quad = |[1 - y_1(\langle w^*, \phi(x_1)\rangle + d^*)]^+ - [1 - y_2(\langle w^*, \phi(x_2)\rangle + d^*)]^+| \\
&\quad \leq |\langle w^*, \phi(x_1) - \phi(x_2)\rangle| \\
&\quad \leq \|w^*\|_{\mathbb{H}} \sqrt{\langle \phi(x_1) - \phi(x_2), \phi(x_1) - \phi(x_2)\rangle} \\
&\quad \leq \sqrt{f_{\mathbb{H}}(\gamma)/c}.
\end{aligned}$$

Here the last inequality follows from the definition of $f_{\mathbb{H}}$. Hence, the example holds by Theorem 6.

Appendix F: Proof of Example 6

It suffices to show the following lemma, which establishes that loss of Lasso solution is Liptschitz continuous.

Lemma 3 *If $w^*(\mathbf{s})$ is the solution of Lasso given training set $\mathbf{s}$, then*

$$|l(w^*(\mathbf{s}), z_a) - l(w^*(\mathbf{s}), z_b)| \leq \left[\frac{1}{nc}\sum_{i=1}^{n}[s_i^{(y)}]^2 + 1\right]\|z_a - z_b\|_\infty.$$

Proof For succinctness we let $y_i = s_i^{(y)}$, $x_i = s_i^{(x)}$ for $i = 1, \ldots, n$. Similarly, we let $y_a = z_a^{(y)}$, $y_b = z_b^{(y)}$, $x_a = z_a^{(x)}$ and $x_b = z_b^{(x)}$. Since $w^*(\mathbf{s})$ is the solution of Lasso, we have (due to optimality)

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i^\top w^*(\mathbf{s}))^2 + c\|w^*(\mathbf{s})\|_1 \leq \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i^\top 0)^2 + c\|0\|_1 = \frac{1}{n}\sum_{i=1}^{n} {y_i}^2,$$

which implies $\|w^*\|_1 \leq \frac{1}{nc}\sum_{i=1}^{n} {y_i}^2$. Therefore,

$$\begin{aligned}
|l(w^*(\mathbf{s}), z_a) - l(w^*(\mathbf{s}), z_b)| &= ||y_a - w^*(\mathbf{s})x_a| - |y_b - w^*(\mathbf{s})x_b|| \\
&\leq |(y_a - w^*(\mathbf{s})x_a) - (y_b - w^*(\mathbf{s})x_b)| \\
&\leq |y_a - y_b| + \|w^*(\mathbf{s})\|_1\|x_a - x_b\|_\infty \\
&\leq (\|w^*(\mathbf{s})\|_1 + 1)\|z_a - z_b\|_\infty \\
&= \left[\frac{1}{nc}\sum_{i=1}^{n} {y_i}^2 + 1\right]\|z_a - z_b\|_\infty.
\end{aligned}$$

Here the first two inequalities holds from triangular inequality, and the last inequality holds due to $z = (x, y)$. □

Appendix G: Proof of Example 7

To see why the example holds, it suffices to show the following lemma, which establishes that the neural network mentioned is Lipschitz continuous. For simplicity, we write the prediction given $x \in \mathcal{X}$ as $NN(x)$.

Lemma 4 *Fixed α, β, if a 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 the following holds*:

$$|l(A_{\mathbf{s}}, z) - l(A_{\mathbf{s}}, \hat{z})| \leq (1 + \alpha^d \beta^d) \|z - \hat{z}\|_\infty.$$

Proof Let x_i^v and $\hat{x}_i^v$ be the output of the ith unit of the vth layer for samples z and $\hat{z}$ respectively. Let $\mathbf{x}^v$ and $\hat{\mathbf{x}}^v$ be the vector such that the ith elements are x_i^v and $\hat{x}_i^v$ respectively. From $\sum_{i=1}^{N_v} |w_i^v| \leq \alpha$ we have

$$\begin{aligned}
|x_i^v - \hat{x}_i^v| &= \left| \sigma\left(\sum_{j=1}^{N_v} w_{ij}^v x_i^{v-1} \right) - \sigma\left(\sum_{j=1}^{N_v} w_{ij}^v \hat{x}_j^{v-1} \right) \right| \\
&\leq \beta \left| \sum_{j=1}^{N_v} w_{ij}^v x_i^{v-1} - \sum_{j=1}^{N_v} w_{ij}^v \hat{x}_j^{v-1} \right| \\
&\leq \beta\alpha \|\mathbf{x}^{v-1} - \hat{\mathbf{x}}^{v-1}\|_\infty.
\end{aligned}$$

Here, the first inequality holds from the Lipschitz condition of σ, and the second inequality holds from $\sum_{j=1}^{N_v} |w_{ij}^v| \leq \alpha$. Iterating over d layers, we have

$$|NN(z^{(x)}) - NN(\hat{z}^{(x)})| = |x^d - \hat{x}^d| \leq \alpha^d \beta^d \|\mathbf{x} - \hat{\mathbf{x}}\|_\infty,$$

which implies

$$\begin{aligned}
|l(A_{\mathbf{s}}, z) - l(A_{\mathbf{s}}, \hat{z})| &= ||z^{(y)} - NN(z^{(x)})| - |\hat{z}^{(y)} - NN(\hat{z}^{(x)})|| \\
&\leq \|z^{(y)} - \hat{z}^{(y)}| + |NN(z^{(x)}) - NN(\hat{z}^{(x)})| \\
&\leq (1 + \alpha^d \beta^d) \|z - \hat{z}\|_\infty.
\end{aligned}$$

This proves the lemma. □

Appendix H: Proof of Example 8

We show that the loss to PCA is Lipschitz continuous, and then apply Theorem 6.

Let $(w_1^*(\mathbf{s}), \ldots, w_d^*(\mathbf{s}))$ be the solution of PCA trained on $\mathbf{s}$. Thus we have

$$\begin{aligned}
&|l((w_1^*(\mathbf{s}), \ldots, w_d^*(\mathbf{s})), z_a) - l((w_1^*(\mathbf{s}), \ldots, w_d^*(\mathbf{s})), z_b)| \\
&= \left| \sum_{k=1}^{d} (w_k^*(\mathbf{s})^\top z_a)^2 - \sum_{k=1}^{d} (w_k^*(\mathbf{s})^\top z_b)^2 \right| \\
&\leq \sum_{k=1}^{d} |[w_k^*(\mathbf{s})^\top z_a - w_k^*(\mathbf{s})^\top z_b][w_k^*(\mathbf{s})^\top z_a + w_k^*(\mathbf{s})^\top z_b]| \leq 2dB\|z_a - z_b\|_2,
\end{aligned}$$

where the last inequality holds because $\|w_k^*(\mathbf{s})\|_2 = 1$ and $\|z_a\|, \|z_b\| \leq B$. Hence, the example holds by Theorem 6.

Appendix I: Proof of Example 9

Let $c_1, \ldots, c_{\mathcal{N}(\gamma/2,\mathcal{X},\rho)}$ be a $\gamma/2$ cover of $\mathcal{X}$. Thus, we can partition $\mathcal{Z}$ to $2\mathcal{N}(\gamma/2, \mathcal{X}, \rho)$ subsets $\{C_i\}$, such that if

$$z_1, z_2 \in C_i; \quad \Longrightarrow \quad y_1 = y_2; \ \& \rho(x_1, x_2) \leq \gamma.$$

Consider an arbitrary $\mathbf{s} \in \mathcal{Z}^n$ and set $\hat{\mathbf{s}}$ as

$$\hat{\mathbf{s}} \triangleq \{s_i \in \mathbf{s} | \mathcal{D}(s_i, \mathcal{A}_\mathbf{s}) > \gamma\}.$$

We then have $|\hat{\mathbf{s}}| = \hat{n}(\mathbf{s})$, and

$$z_1 \in \hat{\mathbf{s}}, \ z_1, z_2 \in C_i; \quad \Longrightarrow \quad y_1 = y_2; \ \mathcal{A}_\mathbf{s}(x_1) = \mathcal{A}_\mathbf{s}(x_2); \quad \Longrightarrow \quad l(\mathcal{A}_\mathbf{s}, z_1) = l(\mathcal{A}_\mathbf{s}, z_2).$$

By definition, $\mathcal{A}$ is $(2\mathcal{N}(\gamma/2, \mathcal{X}, \rho), 0, \hat{n}(\cdot))$ pseudo robust.

References

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

Bartlett, P. L. (1998). 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.

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

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

Ben-David, S., Blitzer, J., Crammer, K., & Pereira, F. (2007). Analysis of representations for domain adaptation. In *Advances in neural information processing systems* (Vol. 19).

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

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

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

Bhattacharyya, C., Pannagadatta, K. S., & Smola, A. J. (2004). A second order cone programming formulation for classifying missing data. In L. K. Saul, Y. Weiss, & L. Bottou (Eds.), *Advances in neural information processing systems (NIPS17)*. Cambridge: MIT Press.

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

Bousquet, O., Boucheron, S., & Lugosi, G. (2005). Theory of classification: a survey of recent advances. *ESAIM Probability and Statistics*, *9*, 323–375.

Cade, B., & Noon, B. (2003). A gentle introduction to quantile regression for ecologists. *Frontiers in Ecology and the Environment*, *1*, 412–420.

Caramanis, C., Mannor, S., & Xu, H. (2011). Robust optimization and machine learning. In S. Sra, S. Nowozin, & S. Wright (Eds.), *Optimization for machine learning*. Cambridge: MIT Press.

Cole, T. (1988). Fitting smoothed centile curves to reference data. *Journal of the Royal Statistical Society, Ser. A*, *151*, 385–418.

Cortes, C., & Vapnik, V. N. (1995). Support vector networks. *Machine Learning*, *20*, 1–25.

Devroye, L., & Wagner, T. (1979a). Distribution-free inequalities for the deleted and holdout error estimates. *IEEE Transactions of Information Theory*, *25*(2), 202–207.

Devroye, L., & Wagner, T. (1979b). Distribution-free performance bounds for potential function rules. *IEEE Transactions of Information Theory*, *25*(2), 601–604.
Devroye, L., Györfi, L., & Lugosi, G. (1996). *A probabilistic theory of pattern recognition*. New York: Springer.
Doob, J. L. (1953). *Stochastic processes*. New York: Wiley.
Evgeniou, T., Pontil, M., & Poggio, T. (2000). Regularization networks and support vector machines. In A. J. Smola, P. L. Bartlett, B. Schölkopf, & D. Schuurmans (Eds.), *Advances in large margin classifiers* (pp. 171–203). Cambridge: MIT Press.
Gamarnik, D. (2003). Extension of the PAC framework to finite and countable Markov chains. *IEEE Transaction on Information Theory*, *49*(1), 338–345.
Globerson, A., & Roweis, S. (2006). Nightmare at test time: robust learning by feature deletion. In *Proceedings of the 23rd international conference on machine learning* (pp. 353–360). New York: ACM Press.
Glynn, P. W., & Ormoneit, D. (2002). Hoeffding's inequality for uniformly ergodic Markov chains. *Statistics and Probability Letters*, *56*, 143–146.
Huber, P. J. (1981). *Robust statistics*. New York: Wiley.
Kakade, S., Sridharan, K., & Tewari, A. (2009). On the complexity of linear predictions: risk bounds, margin bounds, and regularization. In *Advances in neural information processing systems* (Vol. 21, pp. 793–800).
Kearns, M., & Schapire, R. (1994). Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and System Sciences*, *48*(3), 464–497.
Klivans, A., Long, P., & Servedio, R. (2009). Learning halfspaces with malicious noise. *Journal of Machine Learning Research*, *10*, 2715–2740.
Koenker, R., & Bassett, G. (1978). Regression quantiles. *Econometrica*, *46*, 33–50.
Kolmogorov, A. N., & Tihomirov, V. (2002). ϵ-entropy and ϵ-capaciity of sets in functional spaces. *American Mathematical Society Translations (2)*, *17*, 227–364.
Koltchinskii, V. (2002). Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, *47*(5), 1902–1914.
Koltchinskii, V., & Panchenko, D. (2002). Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics*, *30*(1), 1–50.
Lanckriet, G. R., El Ghaoui, L., Bhattacharyya, C., & Jordan, M. I. (2003). A robust minimax approach to classification. *Journal of Machine Learning Research*, *3*, 555–582.
Lozano, A. C., Kulkarni, S. R., & Schapire, R. E. (2006). Convergence and consistency of regularized boosting algorithms with stationary β-mixing observations. In *Advances in neural information processing systems* (Vol. 18).
Mansour, Y., Mohri, M., & Rostamizadeh, A. (2009). Domain adaptation: learning bounds and algorithms. In *Proceedings of the 22nd annual conference on learning theory*.
McDiarmid, C. (1989). On the method of bounded differences. In *Surveys in combinatorics* (pp. 148–188).
Meyn, S. P., & Tweedie, R. L. (1993). *Markov chains and stochastic stability*. New York: Springer.
Mukherjee, S., Niyogi, P., Poggio, T., & Rifkin, R. (2006). 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.
Poggio, T., Rifkin, R., Mukherjee, S., & Niyogi, P. (2004). General conditions for predictivity in learning theory. *Nature*, *428*(6981), 419–422.
Rousseeuw, P. J., & Leroy, A. M. (1987). *Robust regression and outlier detection*. New York: Wiley.
Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels*. Cambridge: MIT Press.
Shalev-Shwartz, S., Shamir, O., Srebro, N., & Sridharan, K. (2009). Learnability and stability in the general learning setting. In *Proceedings of 22nd annual conference of learning theory*.
Shivaswamy, P. K., Bhattacharyya, C., & Smola, A. J. (2006). Second order cone programming approaches for handling missing and uncertain data. *Journal of Machine Learning Research*, *7*, 1283–1314.
Steinwart, I. (2006). Consistency of support vector machines and other regularized kernel classifiers. *IEEE Transactions on Information Theory*, *51*(1), 128–142.
Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: an introduction*. Cambridge: MIT Press.
Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, *58*(1), 267–288.
van der Vaart, A. W., & Wellner, J. A. (2000). *Weak convergence and empirical processes*. New York: Springer.
Vapnik, V. N., & Chervonenkis, A. (1974). *Theory of pattern recognition*. Moscow: Nauka.
Vapnik, V. N., & Chervonenkis, A. (1991). The necessary and sufficient conditions for consistency in the empirical risk minimization method. *Pattern Recognition and Image Analysis*, *1*(3), 260–284.
Xu, H., & Mannor, S. (2010). Robustness and generalization. In *Proceedings of the twenty-third annual conference on learning theory* (pp. 503–515).

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

Xu, H., Caramanis, C., Mannor, S., & Yun, S. (2009b). Risk sensitive robust support vector machines. In *Proceedings of the forty-eighth IEEE conference on decision and control* (pp. 4655–4661).

Xu, H., Caramanis, C., & Mannor, S. (2010a). Principal component analysis with contaminated data: the high dimensional case. In *Proceeding of the twenty-third annual conference on learning theory* (pp. 490–502).

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

Yu, Y., Yang, M., Xu, L., White, M., & Schuurmans, D. (2011). Relaxed clipping: a global training method for robust regression and classification. In *Advances in neural information processing systems* (Vol. 23).

Zakai, A., & Ritov, R. (2009). Consistency and localizability. *Journal of Machine Learning Research*, *10*, 827–856.

Zou, B., Li, L. Q., & Xu, Z. B. (2009). The generalization performance of ERM algorithm with strongly mixing observations. *Machine Learning*, *75*, 275–295.