SIAM J. CONTROL OPTIM.
Vol. 51, No. 5, pp. 3532–3557

RANDOM CONVEX PROGRAMS WITH L_1-REGULARIZATION: SPARSITY AND GENERALIZATION*

M. C. CAMPI† AND A. CARÈ†

Abstract. *Random convex programs* are convex optimization problems that are robust with respect to a finite number of randomly sampled instances of an uncertain variable δ. This paper studies random convex programs in which there is uncertainty in the objective function. Specifically, let $L(x, \delta)$ be a loss function that is convex in x, the optimization variable, while it has an arbitrary dependence on the random variable δ representing uncertainty in the optimization problem. After sampling N instances $\delta^{(1)}, \delta^{(2)}, \ldots, \delta^{(N)}$ of the random variable δ, the random convex program can be written as follows: $\min_x \max_i L(x, \delta^{(i)})$. The fundamental feature of this program is that its value $L_N^* = \max_i L(x_N^*, \delta^{(i)})$, where x_N^* is the solution, remains guaranteed when x_N^* is applied to the vast majority of the other unseen instances of δ; that is, $L(x_N^*, \delta) \leq L_N^*$ holds with high probability with respect to the uncertain variable δ. This *generalization property* has justified a systematic and rigorous use of randomization in robust optimization. In this paper, we introduce L_1-regularization in random convex programs and show that L_1-regularization boosts the above generalization property so that generalization is achieved with significantly fewer samples than in the standard convex program given above. Explicit bounds are derived that allow a rigorous and easy implementation of the method.

Key words. random programs, L_1-regularization, robustness, sparsity, convex optimization, scenario optimization

AMS subject classifications. 90C15, 90C34, 90C25

DOI. 10.1137/110856204

1. Introduction.

Consider a loss function $L(x, \delta)$, where $x \in \mathcal{X} \subseteq \mathbb{R}^d$ is the optimization variable and $\delta \in \Delta$ is a random variable that describes uncertainty in the optimization problem. Often, set Δ has infinite cardinality. The following convexity assumption is in effect throughout the paper.

ASSUMPTION 1. *Function $L(x, \delta)$ is convex in x, while it has an arbitrary dependence on δ, and the optimization domain $\mathcal{X}$ is a convex and closed set.*

A random convex program [5, 28, 34, 1] is obtained by sampling a *finite number* of δ's from Δ in an i.i.d. (independent and identically distributed) fashion according to the probability distribution $\mathbb{P}$ of δ (these random samples are indicated as $\delta^{(i)}$, $i = 1, \ldots, N$, and called "scenarios"), and by taking worst-case minimization with respect to the scenarios $\delta^{(i)}$, namely,

$$\min_{x \in \mathcal{X} \subseteq \mathbb{R}^d} \max_{i=1,\ldots,N} L(x, \delta^{(i)}).^1 \tag{1.1}$$

Problem (1.1) is a convex program where the function to be minimized is obtained as the max of N convex functions $L(x, \delta^{(i)})$ and can be practically solved via standard

*Received by the editors November 22, 2011; accepted for publication (in revised form) June 17, 2013; published electronically September 19, 2013. This research was supported by MIUR (Ministero dell'Istruzione, dell'Università e della Ricerca).

http://www.siam.org/journals/sicon/51-5/85620.html

†Dipartimento di Ingegneria dell'Informazione, Università di Brescia, 25123 Brescia, Italia (marco.campi@ing.unibs.it, http://www.ing.unibs.it/campi/; algo.care@ing.unibs.it, http://www.ing.unibs.it/algo.care/).

[1]Depending on the application, the scenarios $\delta^{(i)}$ can be obtained either from a probabilistic model or from actual observations. In the former, Δ and $\mathbb{P}$ are part of the model specification, and the $\delta^{(i)}$'s are attained by random computer generation. In the latter, the $\delta^{(i)}$'s are observed, and the $\delta^{(i)}$'s are assumed to have a common distribution $\mathbb{P}$.

optimization programs, such as `CVX` [21, 22] or `YALMIP` [27]. The crucial feature of (1.1) is that its solution comes accompanied by precise theoretical guarantees that relate the solution of (1.1) to other instances of the uncertainty parameter δ that do not appear explicitly in problem (1.1) [5, 7]. Indeed, the optimal value of (1.1) $L_N^* = \max_{i=1,\dots,N} L(x_N^*, \delta^{(i)})$, where x_N^* is the optimal solution of (1.1), is a guaranteed cost for the vast majority of the unseen instances δ, up to a probabilistic level ϵ that the user can specify before running program (1.1). This *generalization property* can be formally stated as follows.

GENERALIZATION PROPERTY 1. *There is a set* Δ_ϵ *with* $\mathbb{P}\{\Delta_\epsilon\} \geq 1-\epsilon$ *such that* $\max_{\delta\in\Delta_\epsilon} L(x_N^*, \delta) \leq L_N^*$.

Thus, generalization in this context is interpreted as *performance robustness*, and the theory of [5, 7] proves that random convex programs represent a viable and systematic approach to obtaining solutions carrying a prescribed performance robustness level ϵ. The reader is referred to [5, 7] for a precise statement of these results. In [6], random convex programs are applied to control problems, and [20] considers solutions where max in (1.1) applies to a subset of the sampled scenarios. Moreover, [5, 6, 7, 20] provide a broad discussion on the relation between (1.1) and deterministic robust programs [30, 3, 4].

The use of L_1-regularization. The number N of samples $\delta^{(i)}$ that have to be drawn to achieve a desired robustness level ϵ increases with the number of optimization variables d, and in practice may result in *too many samples* in applications where the number of variables is large [31, 34]. The central focus of this paper is on how this critical obstacle can be alleviated by L_1*-regularization*. L_1-regularization is employed to reduce the effective dimension of the optimization variable, and this reduction allows us to find robust solutions with substantially fewer samples.

L_1-regularization was discussed in [37] for regression problems, and since then L_1-regularization has been used by many in model fitting [25, 38, 26, 16, 10, 29, 42, 12, 44], as well as in signal processing, where this approach is also known under the name *variable pursuit* [13, 17, 9, 8]. Moreover, L_1-optimization has given rise to the emerging field of compressed sensing; see, e.g., [15, 11]. In [42], it has been shown that in a regression context L_1-regularization is intimately tied to robust optimization, a fact that holds more generally for other types of regularization as well [41, 43]. Building on this connection, in [44] bounds on the difference between the expected error and the average training error have been derived. These bounds are generalization results in a statistical learning sense. Reference [12] gives a nice overview of these findings. The present paper introduces L_1-regularization in random convex programs and provides a rigorous theory for establishing conditions under which the previously introduced Generalization Property 1 holds. Our results are inherently different from those in [44] in two respects. First, generalization is here intended as the ability of L_N^* to be an upper bound to $L(x_N^*, \delta)$ with high probability, as opposed to the average statistical learning sense of [44]. Second, the analysis of [44] hinges upon the concept of *algorithmic robustness*; i.e., the algorithm achieves similar performance for testing samples that are close to the training samples. Here, δ has no other structure than being a random variable, and the concept itself that δ_1 and δ_2 are close to each other has no meaning. Our results are obtained in the spirit of [5, 7], where it is shown that generalization holds provided the number of scenarios $\delta^{(i)}$ that determine the solution x_N^* (*support scenarios*) is small; see [5, 7] for more discussion on this approach and a comparison with other methods.

Structure of the paper. In the next section we formally introduce the setup for random convex optimization with L_1-regularization. The generalization properties of random convex programs with L_1-regularization are studied in section 3, while section 4 discusses the practical use of the method through numerical examples. Section 5 provides some complementary theoretical results.

2. Random convex optimization with L_1-regularization.

2.1. Random convex programs with L_1-regularization. A random convex program with L_1-regularization is written as

$$\boldsymbol{L_1-}\textbf{RCP}: \min_{x\in\mathcal{X}\subseteq\mathbb{R}^d} \max_{i=1,\dots,N} L(x,\delta^{(i)}) \tag{2.1}$$

$$\text{subject to } \|Ax-b\|_1 \le r, \tag{2.2}$$

where, as before, $\delta^{(i)}$, $i=1,\dots,N$, are N scenarios sampled from Δ in an i.i.d. fashion according to $\mathbb{P}$, A is a $p\times d$ matrix, b is a vector of dimension p, $\|\cdot\|_1$ is the 1-norm ($\|z\|_1=\sum_j |z_j|$, where z_j are the components of z), and $r\in\mathbb{R}$ is the "constraining parameter."[2] In many cases of interest, as, e.g., Examples 1 and 2 below, $p=d$. As p moves away from d, the generalization results of section 3 lose strength. More discussion on this point is provided in Remark 3.6 in section 3. Depending on the choice of A and b, (2.2) accommodates constraints of different shape. A couple of examples are now given for concreteness.

Example 1 (lasso constraint). Letting $A=I$ and $b=0$, (2.2) is written as

$$\|x\|_1 \le r. \tag{2.3}$$

Figure 2.1 shows the diamond-shaped form of constraint (2.3) for $d=3$.

The L_1-regularization of x given by (2.3) has been used in regression problems in [37] under the name of lasso regularization, and it has since stimulated a lot of activity. Its main feature is that it has a tendency to return *sparse* solutions, i.e., solutions having a large number of zero components x_j. See [23] for ample discussion of this sparsity effect, and section 3 of this paper for a study of this effect in the specific context of random convex optimization. Sparsity slims down the solution and permits one to gain insight into the operativity of the design. Moreover, sparsity results in solutions that have improved generalization properties, that is, to robust solutions with fewer samples. Section 4 presents simulation examples where the lasso constraint is used. ■

Example 2 (basalt column constraint). Take

$$A=\begin{bmatrix} 1 & -1 & 0 & 0 & \cdots \\ 0 & 1 & -1 & 0 & \cdots \\ & & \vdots & & \\ \cdots & 0 & 0 & 1 & -1 \\ -1 & 0 & \cdots & 0 & 1 \end{bmatrix},$$

[2] A random convex program with L_1-regularization can be rewritten as a standard random convex program by incorporating the L_1-constraint (2.2) in the definition of the optimization domain $\mathcal{X}$. The reason for writing constraint (2.2) explicitly is that in section 2.2 the constraining parameter r is varied and tuned so that a certain level of sparsity is achieved.

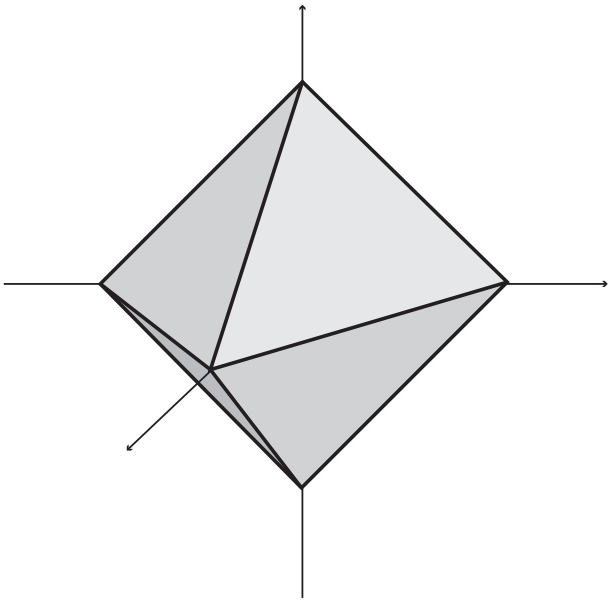

FIG. 2.1. *The lasso constraint.*

and $b = 0$ in (2.2) to obtain

$$\left\| \begin{bmatrix} x_1 - x_2 \\ x_2 - x_3 \\ \vdots \\ x_d - x_1 \end{bmatrix} \right\|_1 \leq r. \tag{2.4}$$

For $d = 3$, the L_1-constraint (2.4) is shaped as a hexagonal basalt column; see Figure 2.2. In this case, the solution has a tendency to set to zero the difference variables $x_j - x_{j+1}$ and $x_d - x_1$ appearing in the constraint (2.4), that is, to favor solutions with equal adjacent components (*sparsity* of the difference variables). This feature can be useful in various applications, such as the optimization of piecewise constant functions or signals where one wishes to moderate the number of jumps [33, 32]. ■

An $\boldsymbol{L_1}$**–RCP** tends to set to zero some of the optimization variables, or linear combinations of them. The selection of A and b is dictated by insight into the problem being solved. Situations often arise where one seeks a sparse solution having many zero components, in which case a lasso constraint is used; see, e.g., the references in the introduction. In other situations, one can be interested in setting to zero certain linearly transformed variables, that is, components of a vector $Ax - b$ as discussed at the end of Example 2. The theory developed in this paper applies to this general case as well. We also note that constraint $\|Ax - b\|_1 \leq r$ is indeed more general than $\|x\|_1 \leq r$, and a problem with a constraint $\|Ax - b\|_1 \leq r$ cannot be reduced to a lasso constraint problem by a change of variables $z = Ax - b$. The reason is that A is not an invertible matrix in general, so that $z = Ax - b$ does not represent a one-to-one transformation between x and z. This is already true for the basalt column constraint where A is singular, and it is clearly true whenever A is a rectangular matrix.

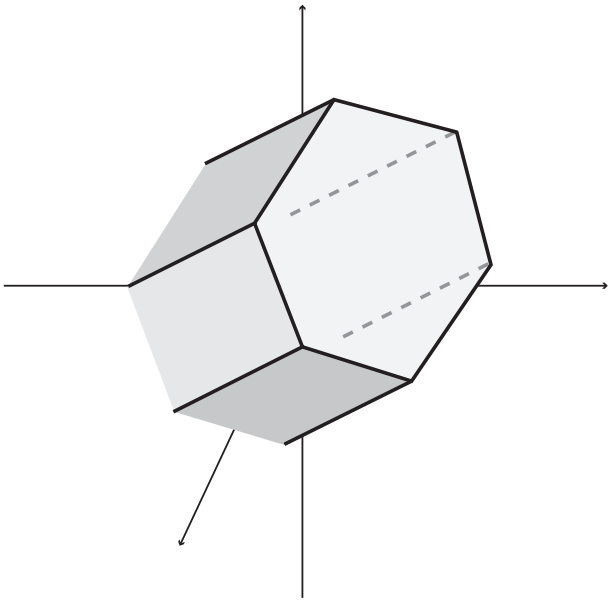

FIG. 2.2. *The basalt column constraint.*

2.2. Random convex algorithm with L_1-regularization. In this section an algorithm is introduced that is able to secure a desired level of generalization.

As the constraining parameter r in (2.2) is increased, the search domain enlarges, the optimal value improves, and the optimal solution loses its generalization properties. In the following algorithm, r is increased until the solution of $\boldsymbol{L_1}$**–RCP** remains confined in a q-dimensional subspace, where q, normally significantly smaller than d, is a user-chosen "complexity parameter" selected before running the optimization algorithm. The generalization properties of the solution stems from the complexity limit set by q, and r is the instrument by means of which the solution is improved until it is empirically observed that the q complexity barrier has been hit.

Random convex algorithm with L_1-regularization ($\boldsymbol{L_1}$–RCA).

(a) Let s be the dimension of the affine subspace of $\mathbb{R}^d$ identified by relation $Ax - b = 0$.[3] Select an integer q with $s < q < d$.
Initialize $r = 0$.[4]

(b) Let $x_N^*(r)$ be the optimal solution path of $\boldsymbol{L_1}$**–RCP** as r is increased.
For all values of $r \geq 0$, evaluate which components of $Ax_N^*(r) - b$ are zero, and let $H(r)$ be the index set of the zero components of $Ax_N^*(r) - b$; thus, if, for example, the first two components of $Ax_N^*(r) - b$ are zero, we have $H(r) = \{1, 2\}$. Further, define $\mathcal{Z}(r) := \{x : a_h^T x - b_h = 0, h \in H(r)\}$, where $a_h^T x - b_h$ is the hth component of $Ax - b$; that is, $\mathcal{Z}(r)$ is the affine subspace of $\mathbb{R}^d$ preserving the null components of $Ax_N^*(r) - b$.

[3] For the lasso constraint of Example 1, $s = 0$, while for the basalt column constraint of Example 2, $s = 1$. Throughout, we assume that $Ax - b = 0$ admits at least one solution.

[4] r is a real parameter that varies continuously over $\mathbb{R}$. For the purpose of running the program, however, r is discretized; see also section 4 on the practical use of $\boldsymbol{L_1}$**–RCA**.

Set $\bar{r}$ to be the largest r such that $\dim(\mathcal{Z}(r)) = q$.[5]

(c) Solve

$$\min_{x \in \mathcal{Z}(\bar{r}) \cap \mathcal{X}} \max_{i=1,\dots,N} L(x, \delta^{(i)}), \tag{2.5}$$

and let x_N^* and L_N^* be the optimal solution and the optimal value of this problem.

Example 1 (continued). For lasso regularization, step (b) prescribes to progressively enlarge the L_1-ball $\|x\|_1 \leq r$. Typically, the number of nonzero components of x increases with r (it is possible that this growth is not monotone; see the example in section 4.1), and one is asked to stop when the optimal solution last switches from q to $q+1$ nonzero components. Optimization in point (c) is performed over the q nonzero components. ■

In general, finding the "optimal" q-dimensional subspace $\mathcal{Z}^{opt}$ determined by setting to zero some of the rows of $Ax-b$ so that $\min_{x \in \mathcal{Z}^{opt} \cap \mathcal{X}} \max_{i=1,\dots,N} L(x, \delta^{(i)}) \leq \min_{x \in \mathcal{Z} \cap \mathcal{X}} \max_{i=1,\dots,N} L(x, \delta^{(i)})$ for any other choice of the q-dimensional subspace $\mathcal{Z}$ is a horrendous combinatorial problem.[6] The L_1-regularization logic implemented in step (b) is a heuristic to find variables that exhibit a strong effect on the optimization cost.

A suitable selection of q has to meet two requirements: guaranteeing adequate generalization properties, while also allowing for a satisfactory optimal cost. In a given application, a priori knowledge on the sparsity of a good solution can be available, and this knowledge can indicate a suitable choice of q. More often, one is driven by empirical evidence. The optimal cost is computed corresponding to various values of q, and a value of q among the tested values is chosen if it meets a satisfying compromise between incurred optimization cost and generalization properties. This a posteriori evaluation procedure can be implemented on a rigorous ground based on the results of this paper, and section 4.2 offers a discussion, along with a numerical example.

In the next section we study the generalization properties of $\boldsymbol{L_1}$**–RCA**. Our ultimate goal is to prove that the user keeps control on the generalization properties by a suitable selection of the complexity parameter q. Numerical results are presented in section 4.

3. Theory: Generalization results. By definition $L_N^* = \max_{i=1,\dots,N} L(x_N^*, \delta^{(i)})$, that is, L_N^* is a guaranteed cost for the scenarios $\delta^{(i)}$. The goal of the present section is to establish the validity of Generalization Property 1 stated in the introduction, that is,

$$\max_{\delta \in \Delta_\epsilon} L(x_N^*, \delta) \leq L_N^* \tag{3.1}$$

holds for a set $\Delta_\epsilon \subseteq \Delta$ that has at least probability $1-\epsilon$. The interpretation is that, when solution x_N^* is applied, cost L_N^* is guaranteed to hold not only for the seen $\delta^{(i)}$'s, but also for most of the unseen situations, those that have not been accounted for

[5] We could have taken "any $\bar{r}$ such that $\dim(\mathcal{Z}(\bar{r})) = q$" instead of the "largest r" and the generalization results in the next section would continue to hold; the only reason for considering the largest r is that the largest r provides better optimization results in normal cases.

[6] A similar problem has been studied in [19] for the case that one wants to minimize the sum of squares, and an efficient algorithm has been derived that works with d values as large as 30.

during the optimization procedure. The precise result is stated below in Theorem 3.2. Theorem 3.2 holds virtually for every $L(x,\delta)$ that are convex in x, so that the theory has wide applicability.

Notation. For future use, we introduce the notation $\boldsymbol{\delta} = (\delta^{(1)}, \delta^{(2)}, \ldots, \delta^{(N)})$ and call $\boldsymbol{\delta}$ a multisample. Note that $\boldsymbol{\delta}$ is a random element in Δ^N, the N-fold Cartesian product of Δ with product probability $\mathbb{P}^N$, where the probability is a product probability because of the independence of the $\delta^{(i)}$'s.

Existence of solutions. We assume existence and uniqueness of the solution to all random convex programs as stated in the next assumption.

ASSUMPTION 2. *With probability* 1 *with respect to the multisample* $\boldsymbol{\delta}$*, any random convex program considered in the analysis of this section admits a unique solution.*

Even though this condition can be relaxed (see, e.g., [5]), we make it because it is not very restrictive and its introduction streamlines the presentation.

Properties of $x^*_N(r)$. $x^*_N(r)$ is a continuous path as a function of r. This is a consequence of the fact that $x^*_N(r)$ is the unique minimizer of a convex function $\max_{i=1,\ldots,N} L(x,\delta^{(i)})$ over a closed domain $\{\|Ax-b\|_1 \le r\} \cap \mathcal{X}$ that expands with continuity as r increases.[7]

For brevity, let $m(r) := \dim(\mathcal{Z}(r))$. $m(r)$ is an integer-valued function that, for each r, returns the dimension of the affine subspace $\mathcal{Z}(r)$ to which the solution $x^*_N(r)$ belongs (refer to point (b) in the $\boldsymbol{L_1}$**–RCA** algorithm). We make the following assumption.

ASSUMPTION 3. *With probability* 1 *with respect to the multisample* $\boldsymbol{\delta}$*, when function* $m(r)$ *increases, it does so one unit at a time, that is, it does not have jumps up of* 2 *or more units, and* $m(\infty) := \lim_{r\to\infty} m(r) = d$.

To see that this assumption is natural, suppose we follow the path $x^*_N(r)$ backward: we start from $r=\infty$ and then progressively shrink the optimization domain by decreasing r. Suppose that, at $r=\tilde{r}$, $m(r)$ drops in value, i.e., $m(\tilde{r}^-) < m(\tilde{r}^+)$; this means that at least one more row $a_h^T x^*_N(\tilde{r}) - b_h$ becomes null. For having a jump down of two or more units, that is, $m(\tilde{r}^-) \le m(\tilde{r}^+) - 2$, $x^*_N(\tilde{r})$ must simultaneously hit two subspaces $a_h^T x - b_h = 0$, which happens only in nongeneric cases. On the other hand, for $r=\infty$ there is no L_1-regularization, so that the optimal solution falls exactly on a subspace $a_h^T x - b_h = 0$, and therefore $m(\infty) < d$, only in nongeneric cases.

Termination of $\boldsymbol{L_1}$–RCA. For $r=0$, $\|Ax^*_N(0)-b\|_1 = 0$ so that $Ax^*_N(0)-b=0$, which entails that $m(0)=s$. Thus, under Assumption 3, $m(r)$ goes from s to d and, when it increases, it does so one unit at a time. Hence, an r exists where $m(r)=q$. Moreover, the sup of the r values for which $m(r)=q$ is indeed a max as it can be argued from the fact that $x^*_N(r)$ is a continuous path, and so $\bar{r}$ in point (b) of $\boldsymbol{L_1}$**–RCA** exists. After $\bar{r}$ is determined in point (b), solving (2.5) at point (c) of $\boldsymbol{L_1}$**–RCA** generates x^*_N and L^*_N and terminates the algorithm. We have proven the following theorem.

THEOREM 3.1. *Under Assumptions* 2 *and* 3*, with probability* 1 *with respect to the multisample* $\boldsymbol{\delta}$*, the* $\boldsymbol{L_1}$**–RCA** *algorithm comes to termination.*

Generalization result. For a multisample $\boldsymbol{\delta}$, $\boldsymbol{L_1}$**–RCA** generates x^*_N. Thus, x^*_N depends on $\boldsymbol{\delta}$, a fact that we henceforth explicitly indicate by the notation $x^*_N(\boldsymbol{\delta})$.

[7] In [35], it is shown that $x^*_N(r)$ is piecewise linear when $L(x,\delta)$ is quadratic in x (in which case $\max_{i=1,\ldots,N} L(x,\delta^{(i)})$ is piecewise quadratic) and $\mathcal{X}$ is a polyhedron so that it has flat faces.

Similarly, we write $L_N^*(\boldsymbol{\delta})$.

Going back to result (3.1), we now see that this result can be more explicitly written as

$$\max_{\delta \in \Delta_\epsilon} L(x_N^*(\boldsymbol{\delta}), \delta) \leq L_N^*(\boldsymbol{\delta}), \tag{3.2}$$

where the appearance of $\boldsymbol{\delta}$ indicates that (3.2) is a random statement. We cannot expect that (3.2) holds true for any multisample $\boldsymbol{\delta}$, as we may stumble upon a multisample that badly represents the variability of δ in Δ. The following theorem asserts a fundamental fact that invalidity of (3.2) happens with a probability that can be made so small as to be negligible for any practical purpose, and this is achieved for reasonable and implementable values of N.

THEOREM 3.2. *Take*

$$N \geq \frac{2}{\epsilon}\left[\ln\frac{1}{\beta} + q + (p-d+q)\ln\left(\frac{p\cdot e}{p-d+q}\right)\right] \tag{3.3}$$

in the $\boldsymbol{L_1}$**–RCA** *algorithm ("ln" is natural logarithm, and "e" is the Nepero constant* $e = 2.718\ldots$*). Under Assumptions* 1, 2, *and* 3, *the following statement holds true with confidence* $1-\beta$, *that is, the statement is true for all multisamples* $\boldsymbol{\delta}$ *with the exception of a set whose probability* $\mathbb{P}^N$ *is at most* β*:*

There is a set Δ_ϵ *with* $\mathbb{P}\{\Delta_\epsilon\} \geq 1-\epsilon$ *such that*

$$\max_{\delta \in \Delta_\epsilon} L(x_N^*(\boldsymbol{\delta}), \delta) \leq L_N^*(\boldsymbol{\delta}). \tag{3.4}$$

Before proving the theorem, some remarks are in order.

REMARK 3.1 (on assumptions in Theorem 3.2). *Theorem* 3.2 *requires Assumptions* 1, 2, *and* 3. *The crucial assumption is the convexity Assumption* 1*; Assumptions* 2 *and* 3 *have a minor role and are introduced only to ensure that the* $\boldsymbol{L_1}$**–RCA** *algorithm comes to termination. Assumptions* 2 *and* 3 *can be easily substituted by other assumptions in modified setups.*

REMARK 3.2 (role of $\mathbb{P}$). *Notice that probability* $\mathbb{P}$ *plays a double role in the theorem statement. Initially,* N *scenarios* $\delta^{(i)}$, $i = 1, \ldots, N$, *are sampled according to* $\mathbb{P}$*; then the generalization property refers to sampling another scenario* δ *again according to* $\mathbb{P}$, *and verifying whether* $L(x_N^*(\boldsymbol{\delta}), \delta) \leq L_N^*(\boldsymbol{\delta})$. *What happens if the testing probability and the verification probability do not coincide? While studying this topic is outside the scope of this paper, we note that this issue has been considered in* [18] *in relation to the classical scenario approach, and the authors showed that, if these two probabilities are not too apart in the Prohorov metric, the generalization property is preserved with minor modifications. It is natural to expect that a similar result can be established in the context of the present paper.*

REMARK 3.3 (role of $\boldsymbol{\beta}$ and $\boldsymbol{\epsilon}$). *The confidence parameter* β *appears in* (3.3) *under the sign of logarithm. This fact is important for the practical appeal of the method because one can take* β *so small, e.g.,* $\beta = 10^{-10}$, *that it can practically be neglected and* (3.3) *is virtually always valid. Picking* $\beta = 10^{-10}$, (3.3) *is written as*

$$N \geq \frac{2}{\epsilon}\left[23.1 + q + (p-d+q)\ln\left(\frac{p\cdot e}{p-d+q}\right)\right], \tag{3.5}$$

where 23.1 *is an upper bound to* $\ln(10^{10})$. *The dependence on* ϵ *is instead inversely proportional, and therefore* N *increases relatively fast as* ϵ *approaches* 0. *This fact limits the range of values of* ϵ *to which the approach can be applied.*

REMARK 3.4 (handy formulas). *The sample complexity in* (3.3) *can be simplified in specific cases. Often* $p = d$*; this is, e.g., the case for the lasso regularization of Example* 1 *and the basalt column regularization of Example* 2*. If so, condition* (3.3) *reduces to*

$$N \geq \frac{2}{\epsilon}\left[\ln\frac{1}{\beta} + q\left(1 + \ln\frac{d\cdot e}{q}\right)\right]. \tag{3.6}$$

In the special case when $\beta = 10^{-10}$*,* (3.6) *further reduces to*

$$N \geq \frac{2}{\epsilon}\left[23.1 + q\left(1 + \ln\frac{d\cdot e}{q}\right)\right]. \tag{3.7}$$

The sample complexity provided by formulas (3.5) *and* (3.7) *returns values for* N *that are reasonable for real implementation in many application problems.*

REMARK 3.5 (a more general result). *Equation* (3.3) *is obtained by making* (3.12) *in the proof explicit with respect to* N*. For easy reference,* (3.12) *is repeated here:*

$$\binom{p}{d-q}\sum_{i=0}^{q}\binom{N}{i}\epsilon^i(1-\epsilon)^{N-i} \leq \beta. \tag{3.8}$$

An explicit expression like (3.3) *comes in handy for a quick computation of* N*, and, moreover, it readily shows certain dependencies (e.g.,* N *scales logarithmically in* $1/\beta$*, and linearly in* $1/\epsilon$*). On the other hand, in typical cases a direct use of formula* (3.8) *leads to an* N *smaller than that given by* (3.3) *by a factor of* 2 *or so. Since solving* (3.8) *for* N *can be troublesome in practice, a ready-to-use MATLAB code for solving* (3.8) *for* N *is provided in Appendix* A.1*.*

Proof of Theorem 3.2. In $\boldsymbol{L_1}$**–RCA**, $x_N^*(\boldsymbol{\delta})$ is obtained by solving program

$$\min_{x\in\mathcal{Z}(\bar{r})\cap\mathcal{X}}\max_{i=1,\dots,N} L(x,\delta^{(i)}),$$

which can be equivalently written in epigraphic form as

$$\min_{L\in\mathbb{R},x\in\mathcal{Z}(\bar{r})\cap\mathcal{X}} L \tag{3.9}$$

$$\text{subject to } L(x,\delta^{(i)}) \leq L, \quad i = 1,\dots,N. \tag{3.10}$$

This program has the structure of a scenario program according to the definition in [7] in $q+1$ variables, where 1 accounts for variable L, and q accounts for the variables x confined to belonging to an affine subspace of dimension q.

Theorem 1 in [7] states that the solution of a scenario program violates more than an ϵ-fraction of the unseen constraints with a small probability that is bounded by a Beta distribution. For a given δ, the constraint is written as $L(x,\delta) \leq L$, and, for $x = x_N^*(\boldsymbol{\delta})$ and $L = L_N^*(\boldsymbol{\delta})$, this constraint becomes $L(x_N^*(\boldsymbol{\delta}),\delta) \leq L_N^*(\boldsymbol{\delta})$. Thus, one might hope to apply Theorem 1 in [7] directly in the present context to show that (3.4) holds with high probability. However, one difficulty in applying Theorem 1 in [7] directly is that the latter reference requires that the domain of the optimization problem be completely set in advance, prior to seeing any $\delta^{(i)}$. Here, instead, the optimization domain $\mathcal{Z}(\bar{r})\cap\mathcal{X}$ for x depends on $\delta^{(i)}$ via the construction of $\mathcal{Z}(\bar{r})$ in the $\boldsymbol{L_1}$**–RCA** algorithm. This difficulty can be circumvented by considering all the potential candidates for $\mathcal{Z}(\bar{r})\cap\mathcal{X}$, a route followed in the reasoning below.

$\mathcal{Z}(\bar{r})$ has by construction dimension q. So, referring to step (b) of the algorithm, $Ax_N^*(\bar{r}) - b$ must have at least $d-q$ null components, and $\mathcal{Z}(\bar{r})$ is determined by $d-q$ linearly independent equations of the type $a_h^T x - b_h = 0$. Now, the number of different ways of choosing $d-q$ equations $a_h^T x - b_h = 0$ out of the p rows of $Ax-b=0$ is given by the binomial coefficient $\binom{p}{d-q}$. Thus, applying Theorem 1 in [7], we arrive at the result that

(3.11)
$$\mathbb{P}^N\{\boldsymbol{\delta} : \mathbb{P}\{\delta \in \Delta : L(x_N^*(\boldsymbol{\delta}), \delta) > L_N^*(\boldsymbol{\delta})\} > \epsilon\} \leq \binom{p}{d-q} \sum_{i=0}^{q} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i},$$

where $\sum_{i=0}^{q} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}$ in the right-hand side is the bound in Theorem 1 in [7] that holds for a given domain of optimization, and the term $\binom{p}{d-q}$ accounts for the number of potential optimization domains. The right-hand side of (3.11) keeps control on the probability of "bad" multisamples $\boldsymbol{\delta}$ such that $\mathbb{P}\{\delta \in \Delta : L(x_N^*(\boldsymbol{\delta}), \delta) > L_N^*(\boldsymbol{\delta})\} > \epsilon$. The complementary condition $\mathbb{P}\{\delta \in \Delta : L(x_N^*(\boldsymbol{\delta}), \delta^{(i)}) > L_N^*(\boldsymbol{\delta})\} \leq \epsilon$ can be equivalently written as follows: there is a set Δ_ϵ with $\mathbb{P}\{\Delta_\epsilon\} \geq 1-\epsilon$ such that

$$\max_{\delta \in \Delta_\epsilon} L(x_N^*(\boldsymbol{\delta}), \delta) \leq L_N^*(\boldsymbol{\delta});$$

so what is left to show (compare with the theorem statement) is that, under condition (3.3) on N, the right-hand side of (3.11) is smaller than or equal to β, i.e.,

$$\binom{p}{d-q} \sum_{i=0}^{q} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i} \leq \beta. \tag{3.12}$$

To show (3.12), start from (3.3) and write

$$N \geq \frac{2}{\epsilon}\left[\ln\frac{1}{\beta} + q + (p-d+q)\ln\left(\frac{p\cdot e}{p-d+q}\right)\right]$$

[to ease notation, let $\alpha := p-d+q$]

$$\Rightarrow N \geq \frac{2}{\epsilon}\left[\ln\frac{1}{\beta} + q + \alpha\ln\left(\frac{p\cdot e}{\alpha}\right)\right]$$

[use $\alpha! \geq (\alpha/e)^\alpha$, which implies that the term added at the next row is not positive]

$$\Rightarrow N \geq \frac{2}{\epsilon}\left[\ln\frac{1}{\beta} + q + \alpha\ln\left(\frac{p\cdot e}{\alpha}\right) + \ln\frac{\left(\frac{\alpha}{e}\right)^\alpha}{\alpha!}\right]$$

$$\Rightarrow N \geq \frac{2}{\epsilon}\left[\ln\frac{1}{\beta} + q + \ln\left[\left(\frac{p\cdot e}{\alpha}\right)^\alpha \cdot \frac{\left(\frac{\alpha}{e}\right)^\alpha}{\alpha!}\right]\right]$$

$$\Rightarrow N \geq \frac{2}{\epsilon}\left[\ln\frac{1}{\beta} + q + \ln\frac{p^\alpha}{\alpha!}\right]$$

$$\Rightarrow \frac{1}{2}N\epsilon - q \geq \ln\frac{1}{\beta} + \ln\frac{p^\alpha}{\alpha!}$$

[since $\frac{(N\epsilon-q)^2}{2N\epsilon} \geq \frac{1}{2}N\epsilon - q$]

$$\Rightarrow \frac{(N\epsilon-q)^2}{2N\epsilon} \geq \ln\frac{1}{\beta} + \ln\frac{p^\alpha}{\alpha!}$$

$$\Rightarrow \ln \beta \geq \ln \frac{p^\alpha}{\alpha!} - \frac{(N\epsilon - q)^2}{2N\epsilon}$$
$$\Rightarrow \beta \geq \frac{p^\alpha}{\alpha!} \cdot \exp\left(-\frac{(N\epsilon - q)^2}{2N\epsilon}\right)$$
$$\Rightarrow \beta \geq \binom{p}{p-\alpha} \cdot \exp\left(-\frac{(N\epsilon - q)^2}{2N\epsilon}\right)$$

[apply Chernoff's bound, which states that the exp term is an upper bound to a Binomial tail; see [14] or [40]]

$$\Rightarrow \beta \geq \binom{p}{p-\alpha} \cdot \sum_{i=0}^{q} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i}, \tag{3.13}$$

which is (3.12). This completes the proof. □

REMARK 3.6 (comparison with [7]). *Suppose that the L_1-regularization is not used, and program* (1.1) *is solved. Program* (1.1) *can be written in epigraphic form as*

$$\begin{aligned} &\min_{L\in\mathbb{R}, x\in\mathcal{X}} L \\ &\text{subject to } L(x, \delta^{(i)}) \leq L, \quad i = 1, \ldots, N, \end{aligned} \tag{3.14}$$

and Theorem 1 *in* [7] *can be applied to this problem, leading to*

$$\sum_{i=0}^{d} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i} \leq \beta. \tag{3.15}$$

Moreover, this same evaluation (3.15) *can also be applied to problem* (2.1), (2.2) *for a fixed r. Indeed,* (2.1), (2.2) *can be written in epigraphic form as*

$$\begin{aligned} &\min_{L\in\mathbb{R}, x\in\bar{\mathcal{X}}} L \\ &\text{subject to } L(x, \delta^{(i)}) \leq L, \quad i = 1, \ldots, N, \end{aligned}$$

where $\bar{\mathcal{X}} = \{x \in \mathcal{X} : \|Ax - b\|_1 \leq r\}$ replaces $\mathcal{X}$ in formulation (3.14). *We want to compare formula* (3.15) *with* (3.8) *when $p = d$.*

The fundamental difference between (3.15) *and* (3.8) *is that the summation in* (3.15) *extends till d, while that in* (3.8) *stops at q; if q is significantly smaller than d, this implies a substantial saving in the sample complexity N. This fact can be appreciated by explicit formulas. By making* (3.15) *explicit with respect to N via a calculation similar to* (3.13), *it is found that*

$$N \geq \frac{2}{\epsilon}\left[\ln \frac{1}{\beta} + d\right].$$

Here N grows linearly with d, while in (3.6) *d is under the sign of logarithm and q replaces the role of d in the linear growth. For a direct numerical comparison of* (3.15) *and* (3.8), *take $\epsilon = 0.2$, $\beta = 10^{-10}$, $p = d = 200$, $q = 7$ (these choices are taken from Example* 1 *in section* 4*). Formula* (3.15) *gives $N = 1469$, while* (3.8) *gives $N = 332$. With the values $\epsilon = 2.03\%$, $\beta = 10^{-10}$, $p = d = 2000$, $q = 6$ taken from Example* 2 *in section* 4 *we instead find $N = 113094$ with* (3.15) *and $N = 4000$ with* (3.8). *Table* 3.1

TABLE 3.1
Values for N obtained using formula (3.15) *(1st line in italic) and formula* (3.8) *(2nd through 9th lines); $\beta = 10^{-10}$, $p = d = 2000$.*

	$\epsilon = 1\%$	$\epsilon = 2\%$	$\epsilon = 3\%$	$\epsilon = 4\%$	$\epsilon = 5\%$	$\epsilon = 6\%$	$\epsilon = 7\%$	$\epsilon = 8\%$	$\epsilon = 9\%$	$\epsilon = 10\%$
	229735	*114793*	*76478*	*57321*	*45826*	*38163*	*32689*	*28584*	*25390*	*22836*
$q = 1$	3403	1693	1123	838	668	554	472	411	364	325
$q = 2$	4427	2203	1462	1091	869	720	614	535	473	424
$q = 3$	5403	2689	1784	1332	1060	879	750	653	578	517
$q = 4$	6346	3158	2096	1564	1245	1033	881	767	679	608
$q = 5$	7264	3615	2399	1791	1426	1182	1009	878	777	696
$q = 10$	11594	5771	3829	2859	2276	1888	1610	1402	1240	1111
$q = 15$	15644	7786	5167	3858	3072	2548	2173	1893	1674	1500
$q = 20$	19506	9709	6443	4810	3831	3177	2711	2361	2088	1870

provides a numerical comparison of (3.15) *and* (3.8) *for various values of ϵ and q. This benefit of having a reduced sample complexity comes from exploiting in Theorem* 3.2 *the structure provided by the sparsity.*

Suppose now that $p \neq d$. L_1-regularization tends to set to zero the rows of $Ax - b$. However, if p is significantly less than d, regularization is not effective and the benefits in terms of sample complexity are lost. This can also be seen in the $\boldsymbol{L_1}$**–RCA** *algorithm where s, the dimension of the affine subspace in $\mathbb{R}^d$ identified by the relation $Ax - b = 0$, is certainly no smaller than $d - p$, i.e., $s \geq d - p$. Since q is bigger than s, we have $q > s \geq d - p$. From this we see that the complexity parameter q cannot be made small compared to d if p is small compared to d. On the other hand, increasing p so that it is much bigger than d results in too many choices of subspaces of dimension q, and this kills the benefit of regularization as well. This is seen from* (3.3)*, where a p as large as $p = 2d$ leads to an N that scales linearly in d. Thus, putting this all together, to exploit the benefit of regularization p must be not too different from d, and $p = d$ is indeed the most common choice.*

4. Practical use of $\boldsymbol{L_1}$–RCA. Section 4.1 presents an example in regression that illustrates the general theory developed in previous sections. Often, a suitable selection of q is made based on empirical evidence, and this aspect is discussed in the example of section 4.2.

4.1. Example 1: Minimax regression. A signal $s(t)$ is obtained as the composition of 200 sinusoids,

$$s(t) = \sum_{j=1}^{200} \alpha_j \sin(jt),$$

where $\alpha_1 = \alpha_5 = \alpha_8 = \alpha_{45} = 0.2$ and the other 196 coefficients are given by the coordinates of a point selected at random on the simplex of size 0.2 (i.e., $\sum_{j \neq 1,5,8,45} \alpha_j = 0.2$, $\alpha_j \geq 0$). Note that $\sum_{j=1}^{200} \alpha_j = 1$ and $\alpha_j \geq 0$, $j = 1, 2, \ldots, 200$; that is, signal $s(t)$ is obtained as the convex combination of 200 sinusoidal waveforms, while just 4 sinusoids are the dominating ones.

We want to construct a reduced order signal $\hat{s}(t)$ that approximates $s(t)$ according to a minimax criterion of best fit. To this purpose, suppose that we can access signal

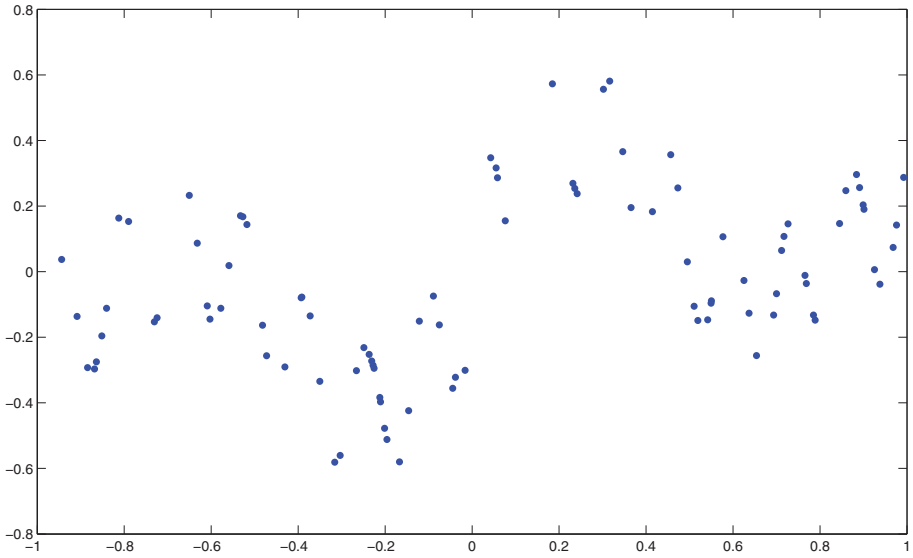

FIG. 4.1. *Samples* $(t^{(i)}, s(t^{(i)}))$.

$s(t)$ by computing its value in correspondence of selected values of the variable t.[8]

To be concrete, suppose $N = 332$ samples $(t^{(i)}, s(t^{(i)}))$ are gathered, where the $t^{(i)}$ are picked independently of each other according to the uniform distribution in $[-\pi, \pi]$. Some of these samples are shown in Figure 4.1. Also, take $\hat{s}(t)$ of the form $\hat{s}(t) = \sum_{k=1}^{7} x_{j_k} \sin(j_k t)$, that is, $\hat{s}(t)$ is composed by 7 sinusoidal waveforms whose frequencies, however, are not a priori decided and instead must be chosen based on the samples. To estimate the 7 frequencies j_k and the associated coefficients x_{j_k}, the $\boldsymbol{L_1}$**–RCA** algorithm is used. First, allow $\hat{s}(t)$ to be formed by 200 sinusoids, that is,

$$\hat{s}(t) = \sum_{j=1}^{200} x_j \sin(jt),$$

and write the $\boldsymbol{L_1}$**–RCP** program (2.1), (2.2) with lasso regularization

$$\min_{x \in \mathbb{R}^{200}} \max_{i=1,\ldots,332} |s(t^{(i)}) - \hat{s}(t^{(i)})|,$$
$$\text{subject to } \|x\|_1 \le r.$$

This program produces solutions $\hat{s}(t)$ of variable complexity, i.e., with a variable number of null coefficients, depending on the regularization parameter r. Figure 4.2 visualizes the number of nonzero coefficients that we obtained when this program was

[8] Even though the simple example of this section has just illustrative purposes, the problem of extracting reduced order descriptions of signals and functions is a fundamental problem in signal processing. Among many contributions, refer, e.g., to [24, 2]. One common assumption of many approximation schemes is that a good approximating function is representable as the linear combination of not too many basis functions, provided one is able to select suitable basis functions from a set of potential candidates that contains many elements. Though very simple, our example here where 4 sinusoids are dominant in a set of 200 elements follows this scheme.

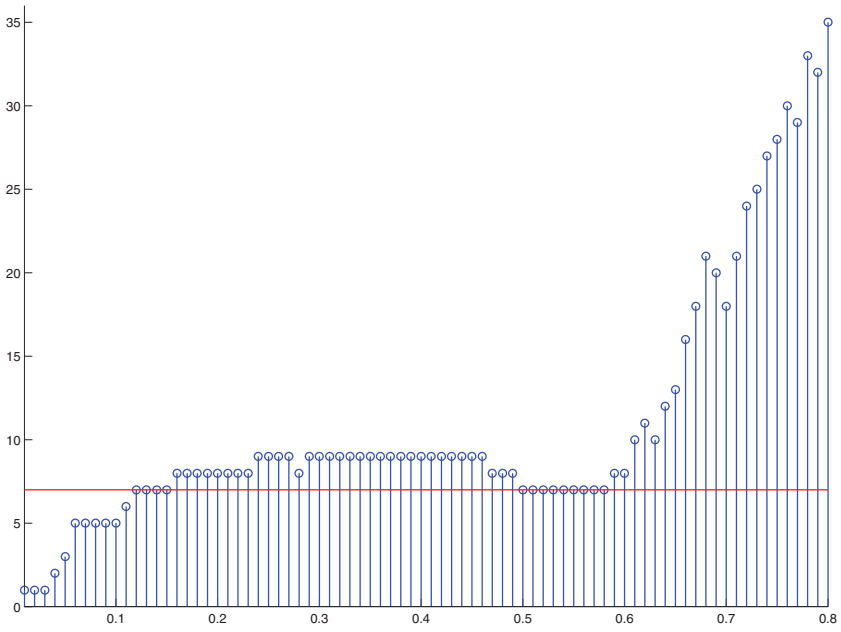

FIG. 4.2. *Number of nonzero coefficients for $r \leq 0.8$; the horizontal line is at level $q = 7$.*

run with the data shown in Figure 4.1, and with r ranging from 0 to 0.8 over a grid with step 0.01.[9]

Selecting $q = 7$ in $\boldsymbol{L_1}$–**RCA**, we halted increasing r at the value $\bar{r} = 0.58$, where the solution had $q = 7$ nonzero coefficients for the last time; the nonzero coefficients were associated with frequencies $j_1 = 1, j_2 = 5, j_3 = 8, j_4 = 41, j_5 = 45, j_6 = 109, j_7 = 127$, showing the ability of the algorithm to capture the sinusoids that have the strongest content in $s(t)$.[10] We further solved

$$\min_{x_{j_1},\ldots,x_{j_7}} \max_{i=1,\ldots,332} \left| s(t^{(i)}) - \sum_{k=1}^{7} x_{j_k} \sin(j_k t^{(i)}) \right|,$$

as prescribed by point (c) of $\boldsymbol{L_1}$–**RCA**, obtaining

$$x^*_{j_1} = 0.1909, \quad x^*_{j_2} = 0.1964, \quad x^*_{j_3} = 0.2033, \quad x^*_{j_4} = 0.0187, \quad x^*_{j_5} = 0.2059,$$
$$x^*_{j_6} = 0.0271, \quad x^*_{j_7} = 0.0184,$$

and optimal value $L^*_{332} = 0.0649$. A portion of the profile of $\hat{s}^*_{332}(t) = \sum_{k=1}^{7} x^*_{j_k} \sin(j_k t)$ against that of $s(t)$ is shown in Figure 4.3.

Next, we use Theorem 3.2 to assess the robustness properties of the solution. To help the reader link the present example to the general theory developed in previous

[9] The reader may have noticed that the jumps up in this function are at times of 2 or more units, and this may appear to be in contradiction with Assumption 3. It is worthwhile pointing out that this behavior is due to the discretization of r and that a finer discretization leads to jumps up of only one unit.

[10] The reader may be interested to know that we repeated the same experiment with $q = 6, 5, 4$ and found that the algorithm was unable to capture all 4 largest sinusoidal components of $s(t)$. This fact has to be ascribed to the heuristic nature of L^1-regularization.

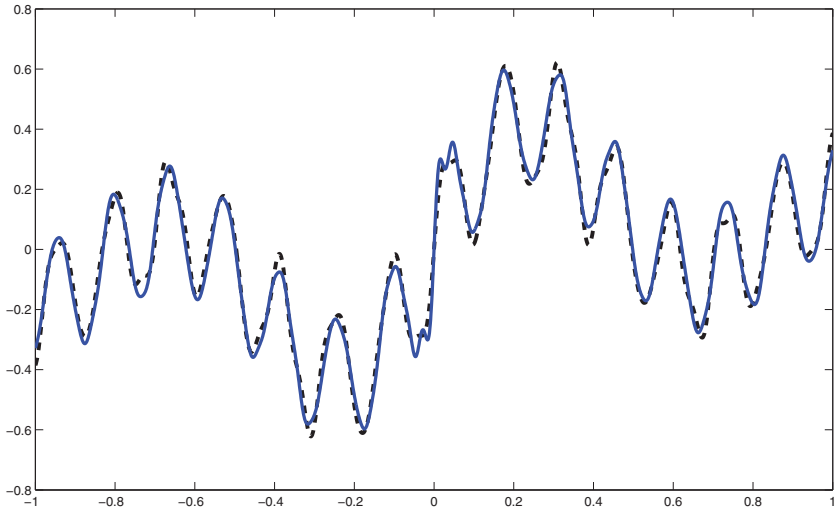

FIG. 4.3. *Signal $s(t)$ (solid line) and reduced order signal $= \hat{s}^*_{332}(t)$ (dashed line).*

sections, we notice that the $t^{(i)}$ here corresponds to the $\delta^{(i)}$ of the general theory, and $[-\pi, \pi]$ is Δ.

Now, substituting $\epsilon = 0.2$, $\beta = 10^{-10}$, $p = 200$, $d = 200$, and $q = 7$ in (3.8), we obtain $N = 332$, which is the actual number of samples we have used. Thereby, an application of Theorem 3.2 permits us to conclude with very high confidence $1-10^{-10}$ that (3.4) holds with $\epsilon = 20\%$, which means that

$$|s(t) - \hat{s}^*_{332}(t)| \leq 0.0649$$

is satisfied with probability at least 80% with respect to random choices of t. Probability 80% can be increased by increasing N. Setting a probability of 95%, from (3.8) we find $N = 1429$, while probability 99% gives $N = 7277$. Increasing N results in an increased computational complexity of the optimization program.

Increasing the confidence in the result by a posteriori evaluation. The value of 80% can also be increased by an a posteriori evaluation that does not involve optimizing over an increased number N of samples. To this end, the following proposition can be applied.

PROPOSITION 4.1. *Let x^*_N be the solution obtained with $\boldsymbol{L_1}$–**RCA**. Take*

$$M \geq \frac{1}{\epsilon'} \ln \frac{1}{\beta'} \tag{4.1}$$

i.i.d. samples $\delta^{(N+1)}, \ldots, \delta^{(N+M)}$ distributed according to $\mathbb{P}$ and independent of $\delta^{(1)}$, $\ldots, \delta^{(N)}$, and let

$$L^* = \max_{i=N+1,\ldots,N+M} L(x^*_N, \delta^{(i)}).$$

Then, with confidence $1 - \beta'$ with respect to the multisample $\delta^{(N+1)}, \ldots, \delta^{(N+M)}$, relation

$$L(x^*_N, \delta) \leq L^* \tag{4.2}$$

holds with probability at least $1 - \epsilon'$ with respect to random choices of δ.

Proof. To prove this simple result, suppose that (4.2) is true with probability less than $1-\epsilon'$, i.e., $L(x_N^*,\delta) \le L^*$ over a set $\Delta_{\epsilon'}$ that has probability $< 1-\epsilon'$. Since, by construction of L^*, $L(x_N^*,\delta^{(i)}) \le L^*$ for all $i = N+1,\dots,N+M$, then all $\delta^{(i)}$'s have to fall in $\Delta_{\epsilon'}$, and the probability of this happening is bounded by $(1-\epsilon')^M$. Imposing that this event is rare and has probability at most β' yields $(1-\epsilon')^M \le \beta'$ or, equivalently, $M \ge \frac{\ln \beta'}{\ln(1-\epsilon')}$. Finally, observe that $\frac{\ln \beta'}{\ln(1-\epsilon')} \le \frac{\ln \beta'}{-\epsilon'} = \frac{\ln \frac{1}{\beta'}}{\epsilon'}$, so that imposing $M \ge \frac{1}{\epsilon'} \ln \frac{1}{\beta'}$ suffices in order that (4.2) holds with probability at least $1-\epsilon'$ with high confidence $1-\beta'$. ∎

For an application of this result to the context of the minimax regression example, pick $\epsilon' = 7\%$, $\beta' = 10^{-10}$, so that $M = 330$. In this case L^* writes $L^* = \max_{i=N+1,\dots,N+330} |s(t^{(i)}) - \hat{s}^*_{332}(t)|$, and we found $L^* = 0.0719$.

The a posteriori evaluation does not require any optimization procedure and can therefore be carried out with little additional computational burden. On the other hand, the a posteriori evaluation does not allow the solution to be adapted to the extra M samples, which are used only for evaluation purposes.

REMARK 4.1 (sample complexity: a priori vs. a posteriori). *The reader has probably noticed that the number $M = 330$ of samples needed for an a posteriori assessment with $\epsilon' = 7\%$ and $\beta' = 10^{-10}$ is similar to the number $N = 332$ of samples needed for applying the* $\boldsymbol{L_1}$**–RCA** *algorithm with $\epsilon = 20\%$ and $\beta = 10^{-10}$. Further inspecting* (3.6) *and* (4.1)*, we see that both of these equations show a linear dependence on the inverse generalization parameters, $1/\epsilon$ or $1/\epsilon'$, and a logarithmic dependence on the inverse confidence parameter, $1/\beta$ or $1/\beta'$. Thus, a posteriori and a priori evaluations have sample complexities that are comparable in structure, a fact that is perhaps surprising.*

REMARK 4.2 (tightness of (3.8)). *One reason that is key to obtaining a result as tight as that given in Theorem* 3.2 *is that in the proof of Theorem* 3.2 *attention is paid to control only the generalization properties in correspondence of x_N^*, the optimal solution. This is in contrast with other generalization theories, chiefly the Vapnik-Chervonenkis theory* [39]*, which aims to control generalization* uniformly *over all potential solutions x.*

REMARK 4.3 (multidimensional functions). *Still referring to the problem of approximating a function from samples, but now broadening our point of view beyond the* 1*-dimensional example of this section, note that the number N of samples given by* (3.8) *does not depend on the dimension of the domain of definition of the function being approximated. Thus, while for the sake of simplicity we have used here an example with a signal of a* 1*-dimensional variable t, should we instead have considered a function defined over a higher-dimensional space, the value of N would have remained the same. This fact is in contrast with deterministic gridding approaches where typically N explodes exponentially with the dimension, a fact known as the "curse of dimensionality."*[11]

4.2. Example 2: Reconstruction of a sparse high-dimensional vector. It may be difficult in a given application to fix the value of q in advance. In these cases, one way to proceed consists in inspecting the optimal value obtained for various values of q, and then selecting a value of q that meets an adequate compromise of performance and robustness guarantees. This procedure is illustrated in this second example.

[11] On the other hand, it is also true that the number of basis functions needed to faithfully reconstruct a function normally scales up with the dimensionality, so that in higher dimensions a larger q is usually selected in $\boldsymbol{L_1}$**–RCA**, and this fact indirectly impacts on the value of N.

A vector z of size 2000 has only 4 nonzero components, whose values are 2.2, -2.7, 1.8, -1.9. z is multiplied by various random vectors $b^{(i)}$, and we are given the result of the multiplication corrupted by an error term $e^{(i)}$. In more precise terms, 4000 vectors $b^{(i)}$ are generated independently of one another. Each $b^{(i)}$ has 2000 components that are uniformly extracted from $[-1, 1]$ and such that each component is independent of the other components. Vector z is multiplied by $b^{(i)}$, thus obtaining $z^T b^{(i)}$, and to the result an error term $e^{(i)}$ is added, whose value is extracted uniformly from $[-0.1, 0.1]$. The error terms $e^{(i)}$ form an independent sequence, which is also independent of the vectors $b^{(i)}$. Thus, our measurements $a^{(i)}$ can be written as

$$a^{(i)} = z^T b^{(i)} + e^{(i)}, \quad i = 1, \ldots, 4000.$$

In order to reconstruct z, we consider the $\boldsymbol{L_1}$**–RCP** program (2.1), (2.2) with lasso regularization

$$\min_{x \in \mathbb{R}^{2000}} \max_{i=1,\ldots,4000} |a^{(i)} - x^T b^{(i)}|$$
$$\text{subject to } \|x\|_1 \le r$$

and set out to solve the corresponding $\boldsymbol{L_1}$**–RCA** algorithm for increasing values of q between 1 and 10. The graph in Figure 4.4 represents the cost obtained for different values of q. Based on this graph, we selected $q = 6$, and this gave a solution x^* where the 4 nonzero components of z were correctly identified up to an error of less than 10^{-4}, while all other components of x^* were less than $3 \cdot 10^{-5}$ in absolute value.[12]

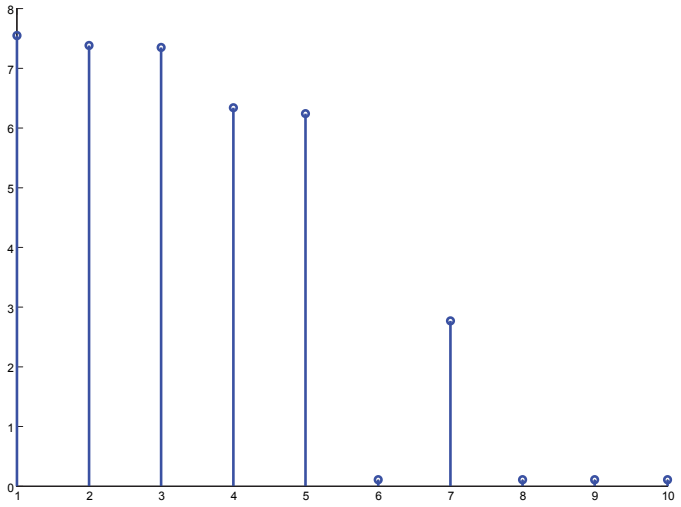

FIG. 4.4. *Cost of* $\boldsymbol{L_1}$**–RCA** *obtained for different values of* q.

Let us now go back to the theory, and see how the theory can be applied to the present context. Formula (3.8) can be applied with $\beta = 10^{-10}$, $p = 2000$, $d = 2000$, $N = 4000$, and $q = 1, 2, \ldots, 10$ obtaining, for each value of q, a different value of ϵ,

[12] For $q = 7$, not all the 4 nonzero components of z were correctly identified. This behavior is due to the heuristic nature of the L_1-regularization procedure. We also notice that a graph like the one shown in Figure 4.4 is sensitive to the selected $b^{(i)}$ and $e^{(i)}$, and for different samples of $b^{(i)}$ and $e^{(i)}$ the graph exhibits a different profile, it can, e.g., be monotonically decreasing.

which we here denote as ϵ_q. q is a posteriori selected based on the cost incurred for different q values, and we denote the selected q as $\bar{q}$. After selecting $\bar{q}$, we want to state that

$$|a - (x^*_{4000})^T b| \leq L^*_{\bar{q}}$$

holds with probability at least $1 - \epsilon_{\bar{q}}$, where $a = z^T b + e$ is a hypothetical new measurement that has not been seen yet, with b and e distributed as $b^{(i)}$ and $e^{(i)}$, but independent of all the seen measurements. Since the statement that

$$|a - (x^*_{4000})^T b| \leq L^*_q \tag{4.3}$$

holds with probability at least $1 - \epsilon_q$ is true for each single $q \in \{1, \ldots, 10\}$ with confidence $1 - 10^{-10}$, the statement that (4.3) holds with probability at least ϵ_q is true for all q *simultaneously* with confidence $1 - 10^{-10} \cdot$ (number of possible choices of q) $= 1 - 10^{-9}$. Therefore, at least with the confidence $1 - 10^{-9}$, we can conclude that when a $\bar{q}$ is a posteriori selected, relation

$$|a - (x^*_{4000})^T b| \leq L^*_{\bar{q}}$$

holds with probability at least $1 - \epsilon_{\bar{q}}$. In our example, we have $\epsilon_{\bar{q}} = \epsilon_6 = 2.03\%$ and $L^*_{\bar{q}} = L^*_6 = 0.0997$, so that the statement that we make with high confidence $1 - 10^{-9}$ is that $|a - (x^*_{4000})^T b| \leq 0.0997$ holds with probability at least 97.97%.

5. An assessment of the robustness-loss curve. $x^*_N(\boldsymbol{\delta})$ denotes the optimal solution obtained by applying the $\boldsymbol{L_1}$**–RCA** algorithm for given N and q. Throughout this section, N and q are kept to fixed values.

Theorem 3.2 establishes that

$$\max_{\delta \in \Delta_\epsilon} L(x^*_N(\boldsymbol{\delta}), \delta) \leq L^*_N(\boldsymbol{\delta}) \tag{5.1}$$

holds over Δ_ϵ with high confidence. This result links a loss value $L^*_N(\boldsymbol{\delta})$ to the probability $1 - \epsilon$ with which such a loss value is guaranteed. The question this section addresses is: Is it possible to go beyond result (5.1) and investigate how rapidly the loss value associated to $x^*_N(\boldsymbol{\delta})$ improves, provided one is ready to decrease the level of probability? We show in this section that a whole robustness-loss curve can in fact be constructed.

Let

$$\epsilon_\ell = \frac{\ell}{N} + \frac{g - 1 + \sqrt{g^2 + 2(\ell - 1)g}}{N}, \quad \ell = q{+}1, \ldots, q{+}h, \tag{5.2}$$

where h is an arbitrary integer chosen by the user such that $q + h \leq N$, and

$$g = \ln \left[\frac{1}{\beta} \cdot \left(\frac{p \cdot e}{\alpha} \right)^\alpha \right], \quad \alpha := p - d + q. \tag{5.3}$$

To ease notation, henceforth we write x^* for $x^*_N(\boldsymbol{\delta})$. Define

$$L^*_{\epsilon_\ell} = \max\{L \text{ such that } L \leq L(x^*, \delta^{(i)}) \text{ for } \ell \text{ scenarios } \delta^{(i)}\}.$$

Thus, $L^*_{\epsilon_\ell}$ are the values $L(x^*, \delta^{(i)})$ listed in decreasing order of magnitude; see Figure 5.1.

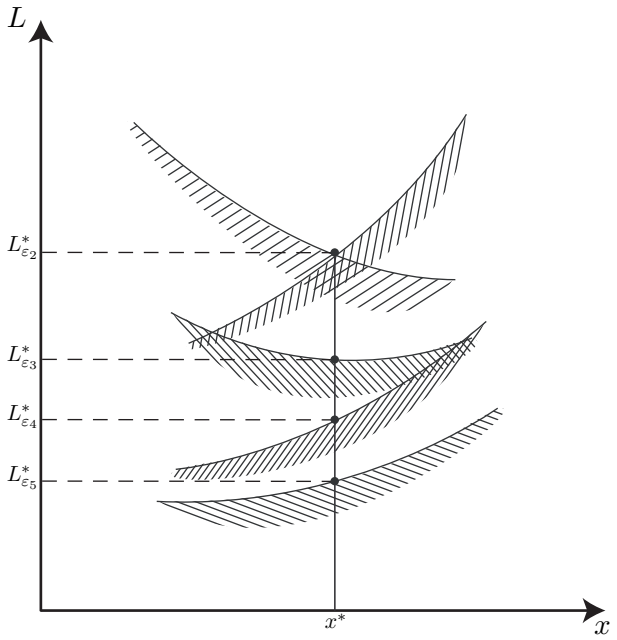

FIG. 5.1. *Visualization of $L^*_{\epsilon_\ell}$ for $q = 1$. Each constraint represents the region where $L \geq L(x, \delta^{(i)})$ for some $\delta^{(i)}$.*

We have the following theorem.

THEOREM 5.1. *The statement*

$$L(x^*, \delta) \leq L^*_{\epsilon_\ell} \quad \textit{holds with probability at least } 1 - \epsilon_\ell$$

is true simultaneously for all $\ell = q+1, \ldots, q+h$ with confidence $1 - h\beta$.

$L^*_{\epsilon_\ell}$ are the loss values corresponding to the scenarios $\delta^{(i)}$'s, and to each $L^*_{\epsilon_\ell}$ the theorem associates a probability $1 - \epsilon_\ell$. Thus, the theorem permits us to determine a robustness-loss curve over the whole range $\ell = q+1, \ldots, q+h$. An example of such a curve is given in Figure 5.3. The proof of the theorem is provided at the end of this section.

REMARK 5.1 (structure of ϵ_ℓ). *ϵ_ℓ is formed by two terms: $\frac{\ell}{N}$ and $\frac{g+\sqrt{g^2+2(\ell-1)g}}{N}$. The first term is the empirical proportion, or the "empirical probability," of the scenarios that are greater than or equal to $L^*_{\epsilon_\ell}$. This empirical proportion alone cannot be expected to be a bound for the real probability that $L(x^*, \delta) > L^*_{\epsilon_\ell}$. Indeed, for one thing an empirical probability is subject to stochastic fluctuation; moreover, in our context, there is a reason of bias for the real probability to be larger than the empirical probability because x^* has been computed via an optimization procedure. The second term $\frac{g+\sqrt{g^2+2(\ell-1)g}}{N}$ is the adjustment term accounting for the mismatch between empirical and real probability. Interestingly, when N is increased and ℓ is kept at a fixed proportion with N, the second term goes to zero as fast as $O(1/\sqrt{N})$.*

REMARK 5.2 (a more general result for ϵ_ℓ). *While* (5.2) *has the advantage of being an explicit expression for ϵ_ℓ, an inspection of the proof of Theorem* 5.1 *reveals*

that the ϵ_ℓ obtained from relation

$$\binom{p}{d-q}\sum_{i=0}^{\ell-1}\binom{N}{i}\epsilon_\ell^i(1-\epsilon_\ell)^{N-i}=\beta \tag{5.4}$$

is still a valid probability for Theorem 5.1 *to hold true. A MATLAB code to compute ϵ_ℓ from* (5.4) *is provided in Appendix* A.2.

Proof of Theorem 5.1. We first concentrate our attention on one ℓ value in the range $[q+1, q+h]$ and bound the probability $\mathbb{P}^N$ of multisamples $\boldsymbol{\delta}$ such that $\mathbb{P}\{L(x^*,\delta) > L^*_{\epsilon_\ell}\} > \epsilon_\ell$.

Similarly to the proof of Theorem 3.2 in section 3, one difficulty is that the optimization domain $\mathcal{Z}(\bar{r})\cap\mathcal{X}$ in point (c) of the algorithm depends on the scenarios $\delta^{(i)}$ via the construction of $\mathcal{Z}(\bar{r})$. We follow the same approach as in the proof of Theorem 3.2 and consider one by one each single candidate domain $\mathcal{Z}(\bar{r})\cap\mathcal{X}$. Correspondingly, in what follows we consider a fixed optimization domain; later on in the proof we shall account for the fact that the optimization domain is one among many candidate domains. This first part of the proof is similar to Part 1 in the proof of Theorem 1 in [7], and is provided here for completeness.

To ease the presentation, we assume that, for any given $(\bar{x},\bar{L})$, $\mathbb{P}\{\bar{L} = L(\bar{x},\delta)\} = 0$. This is a nondegeneracy condition requiring that functions $L = L(x,\delta)$ do not accumulate in any given point $(\bar{x},\bar{L})$; this condition can be removed similarly to Part 2b in the proof of Theorem 1 in [7].

As an intermediate step in the derivation of the final result, we first consider the case when $N = \ell$. Let

$$F(\alpha) := \mathbb{P}^N\{\boldsymbol{\delta} : \mathbb{P}\{L(x^*_\ell,\delta) > L^*_{\epsilon_\ell}\} \le \alpha\} \tag{5.5}$$

be the probability distribution of $\mathbb{P}\{L(x^*_\ell,\delta) > L^*_{\epsilon_\ell}\}$ (here, we write x^*_ℓ to recall that x^* has been obtained with ℓ scenarios). We shall prove that this distribution is

$$F(\alpha) = \alpha^\ell. \tag{5.6}$$

To prove (5.6), consider Δ^m, the space whose elements are m instances of δ, which we write as $(\delta^{(1)},\dots,\delta^{(m)})$. Dimension m is any integer bigger than or equal to ℓ and has to be thought of as a fixed number. Given an element $(\delta^{(1)},\dots,\delta^{(m)})$ of Δ^m, compute the solution to problem (3.9), (3.10), where N is substituted by m, and further single out the indexes of the ℓ functions $L = L(x,\delta^{(i)})$ that are at the top ℓ positions on the line that passes through the solution. For $\ell = 4$ these are the functions that touch the half-line in bold in Figure 5.2. Further, group all elements in Δ^m having the same indexes. In this way, $\binom{m}{\ell}$ sets $S_{\mathcal{I}}$ are constructed forming a partition (up to a probability 0 set) of Δ^m, where $\mathcal{I}\subseteq\{1,\dots,m\}$ is a set of cardinality ℓ containing the indexes of the top ℓ functions. We claim that the probability of each of these sets is

$$\mathbb{P}^m\{S_{\mathcal{I}}\} = \int_0^1 (1-\alpha)^{m-\ell}F(\mathrm{d}\alpha), \tag{5.7}$$

where $F(\alpha)$ is defined in (5.5); using (5.7), later on in the proof we shall show that $F(\alpha)$ has the expression in (5.6).

To establish (5.7) in a more concrete way, consider one of the sets $S_{\mathcal{I}}$, e.g., the set where the indexes of the top ℓ functions are $1,\dots,\ell$. Select fixed values $\bar{\delta}^{(1)},\dots,\bar{\delta}^{(\ell)}$

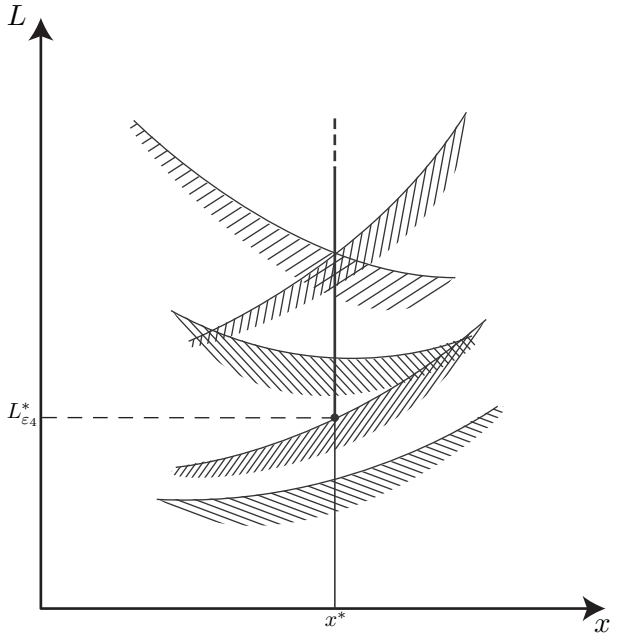

FIG. 5.2. *The 4 functions singled out in the proof of the theorem are those touching the bold half-line.*

for $\delta^{(1)}, \ldots, \delta^{(\ell)}$, solve (3.9), (3.10) with $N = \ell$ for the given $\bar{\delta}^{(1)}, \ldots, \bar{\delta}^{(\ell)}$, and let $\bar{x}^*_\ell$ be the optimal solution and $\bar{L}^*_{\epsilon_\ell}$ the value corresponding to $\bar{x}^*_\ell$ obtained by the lowest function. Let $\alpha(\bar{\delta}^{(1)}, \ldots, \bar{\delta}^{(\ell)}) = \mathbb{P}\{L(\bar{x}^*_\ell, \delta) > \bar{L}^*_{\epsilon_\ell}\}$. Then the probability that $\delta^{(\ell+1)}, \ldots, \delta^{(m)}$ are not among the ℓ top functions, i.e., $(\bar{\delta}^{(1)}, \ldots, \bar{\delta}^{(\ell)}, \delta^{(\ell+1)}, \ldots, \delta^{(m)}) \in S_{\mathcal{I}}$, is $(1 - \alpha(\bar{\delta}^{(1)}, \ldots, \bar{\delta}^{(\ell)}))^{m-\ell}$. Integrating over the domain Δ^ℓ for $(\bar{\delta}^{(1)}, \ldots, \bar{\delta}^{(\ell)})$, we then have

$$\begin{aligned}
&\mathbb{P}^m\{S_{\mathcal{I}}\} \\
&= \int_{\Delta^\ell} (1 - \alpha(\bar{\delta}^{(1)}, \ldots, \bar{\delta}^{(\ell)}))^{m-\ell} \mathbb{P}^\ell(\mathrm{d}\bar{\delta}^{(1)}, \ldots, \mathrm{d}\bar{\delta}^{(\ell)}) \\
&= \int_0^1 (1 - \alpha)^{m-\ell} F(\mathrm{d}\alpha),
\end{aligned}$$

where the second equality is a change of variables from $(\bar{\delta}^{(1)}, \ldots, \bar{\delta}^{(\ell)})$ to α. This establishes (5.7).

Recalling that the sets $S_{\mathcal{I}}$ form a partition of Δ^m up to a probability 0 set and noting that $\mathbb{P}^m\{\Delta^m\} = 1$, (5.7) yields

$$\binom{m}{\ell} \int_0^1 (1 - \alpha)^{m-\ell} F(\mathrm{d}\alpha) = 1 \quad \text{for all } m \geq \ell. \tag{5.8}$$

Expression $F(\alpha) = \alpha^\ell$ in (5.6) is indeed a solution of (5.8), as is easily verified by integration by parts; on the other hand, no other solutions exist, since determining an F satisfying (5.8) is a moment problem for a distribution with finite support and its solution is unique; see, e.g., Corollary 1, section 12.9, Chapter II of [36]. Thus, we have proven that $F(\alpha)$ has the expression (5.6).

Consider next a generic N, not necessarily equal to ℓ. Partition the set $\{\boldsymbol{\delta} : \mathbb{P}\{L(x_N^*, \delta) > L_{\epsilon_\ell}^*\} > \epsilon_\ell\}$ by intersecting it with the $\binom{N}{\ell}$ sets $S_{\mathcal{I}}$ grouping elements of Δ^N such that the ℓ top functions have the same indexes. We then have

$$
\begin{aligned}
&\mathbb{P}^N\{\boldsymbol{\delta} : \mathbb{P}\{L(x_N^*, \delta) > L_{\epsilon_\ell}^*\} > \epsilon_\ell\} \\
&= \mathbb{P}^N\big\{ \cup_{\mathcal{I}} \{\boldsymbol{\delta} : \mathbb{P}\{L(x_N^*, \delta) > L_{\epsilon_\ell}^*\} > \epsilon_\ell\} \text{ and the functions at top} \\
&\qquad \text{positions have indexes in } \mathcal{I}\}\big\} \\
&\quad [\mathbb{I}_A \text{ is the indicator function of set } A, \text{ i.e., } \mathbb{I}_A = 1 \text{ over } A \text{ and } \mathbb{I}_A = 0 \text{ otherwise}] \\
&= \binom{N}{\ell} \int_{\Delta^\ell} (1 - \alpha(\bar{\delta}^{(1)}, \dots, \bar{\delta}^{(\ell)}))^{N-\ell} \mathbb{I}_{\{\alpha(\bar{\delta}^{(1)}, \dots, \bar{\delta}^{(\ell)}) > \epsilon_\ell\}} \mathbb{P}^\ell(\mathrm{d}\bar{\delta}^{(1)}, \dots, \mathrm{d}\bar{\delta}^{(\ell)}) \\
&= \binom{N}{\ell} \int_{\epsilon_\ell}^1 (1 - \alpha)^{N-\ell} \, F(\mathrm{d}\alpha) \\
&\quad [\text{since } F(\mathrm{d}\alpha) = \ell\alpha^{\ell-1} \, \mathrm{d}\alpha] \\
&= \binom{N}{\ell} \int_{\epsilon_\ell}^1 \left[(1 - \alpha)^{N-\ell} \ell\alpha^{\ell-1}\right] \, \mathrm{d}\alpha \\
&\quad [\text{integrating by parts}] \\
&= \binom{N}{\ell} \left[-\frac{(1-\alpha)^{N-\ell+1}}{N-\ell+1} \ell\alpha^{\ell-1} \Big|_{\epsilon_\ell}^1 + \int_{\epsilon_\ell}^1 \frac{(1-\alpha)^{N-\ell+1}}{N-\ell+1} \ell(\ell-1)\alpha^{\ell-2} \, \mathrm{d}\alpha \right] \\
&= \binom{N}{\ell-1} \epsilon_\ell{}^{\ell-1} (1-\epsilon_\ell)^{N-\ell+1} + \binom{N}{\ell-1} \int_{\epsilon_\ell}^1 (1-\alpha)^{N-\ell+1} (\ell-1)\alpha^{\ell-2} \, \mathrm{d}\alpha \\
&= \cdots \\
&= \binom{N}{\ell-1} \epsilon_\ell{}^{\ell-1} (1-\epsilon_\ell)^{N-\ell+1} + \cdots + \binom{N}{1} \epsilon_\ell (1-\epsilon_\ell)^{N-1} + \binom{N}{1} \int_{\epsilon_\ell}^1 (1-\alpha)^{N-1} \, \mathrm{d}\alpha \\
&= \sum_{i=0}^{\ell-1} \binom{N}{i} \epsilon_\ell{}^i (1-\epsilon_\ell)^{N-i}.
\end{aligned}
$$

Next, we sum up over all the potential candidates for $\mathcal{Z}(\bar{r}) \cap \mathcal{X}$, which are $\binom{p}{d-q}$, thus obtaining

$$
\mathbb{P}^N\{\boldsymbol{\delta} : \mathbb{P}\{L(x_N^*, \delta) > L_{\epsilon_\ell}^*\} > \epsilon_\ell\} \le \binom{p}{d-q} \sum_{i=0}^{\ell-1} \binom{N}{i} \epsilon_\ell^i (1-\epsilon_\ell)^{N-i},
$$

or equivalently that

$$
L(x_N^*, \delta) \le L_{\epsilon_\ell}^* \quad \text{holds with probability at least } 1 - \epsilon_\ell \tag{5.9}
$$

with confidence $1 - \binom{p}{d-q} \sum_{i=0}^{\ell-1} \binom{N}{i} \epsilon_\ell^i (1-\epsilon_\ell)^{N-i}$.

When ℓ varies in the range $[q+1, q+h]$, the conclusion is drawn that (5.9) is true simultaneously for all $\ell = q+1, \dots, q+h$ with confidence

$$
1 - \sum_{\ell=q+1}^{q+h} \left[\binom{p}{d-q} \sum_{i=0}^{\ell-1} \binom{N}{i} \epsilon_\ell^i (1-\epsilon_\ell)^{N-i} \right]. \tag{5.10}
$$

If we show that each term in square brackets is bounded by β, namely,

$$
\binom{p}{d-q} \sum_{i=0}^{\ell-1} \binom{N}{i} \epsilon_\ell^i (1-\epsilon_\ell)^{N-i} \le \beta, \qquad \ell = q+1, \dots, q+h, \tag{5.11}
$$

then confidence (5.10) is not less than $1 - h\beta$, and the proof is complete.

To prove (5.11) observe first that the ϵ_ℓ given by (5.2) satisfies relation

$$\frac{(N\epsilon_\ell - \ell + 1)^2}{2N\epsilon_\ell} = g,$$

from which, recalling the expression for g in (5.3), we obtain

$$\left(\frac{p \cdot e}{\alpha}\right)^\alpha \cdot \exp\left(-\frac{(N\epsilon_\ell - \ell + 1)^2}{2N\epsilon_\ell}\right) = \beta. \tag{5.12}$$

The first term on the left-hand side is lower-bounded by

$$\left(\frac{p \cdot e}{\alpha}\right)^\alpha \geq [\text{use } \alpha! \geq (\alpha/e)^\alpha] \geq \frac{p^\alpha}{\alpha!} \geq \binom{p}{p-\alpha} = \binom{p}{d-q}, \tag{5.13}$$

while an application of the Chernoff bound for the Binomial tail (see [14] or [40]) to the second term yields

$$\exp\left(-\frac{(N\epsilon_\ell - \ell + 1)^2}{2N\epsilon_\ell}\right) \geq \sum_{i=0}^{\ell-1} \binom{N}{i} \epsilon_\ell^i (1-\epsilon_\ell)^{N-i}. \tag{5.14}$$

Substituting (5.13) and (5.14) in (5.12) gives (5.11). This concludes the proof. ∎

5.1. Example 1: Minimax regression, continued. Consider again the example of section 4.1 with $N = 7277$ samples. As seen in section 4.1, $N = 7277$ is enough samples to guarantee with confidence $1 - 10^{-10}$ that

$$|s(t) - \hat{s}^*_{7277}(t)| \leq L^*_{7277}$$

holds with probability at least 99%. The value found for L^*_{7277} was $L^*_{7277} = 0.0834$. Figure 5.3 shows the robustness-loss curve where $L^*_{\epsilon_\ell}$ is represented on the vertical axis

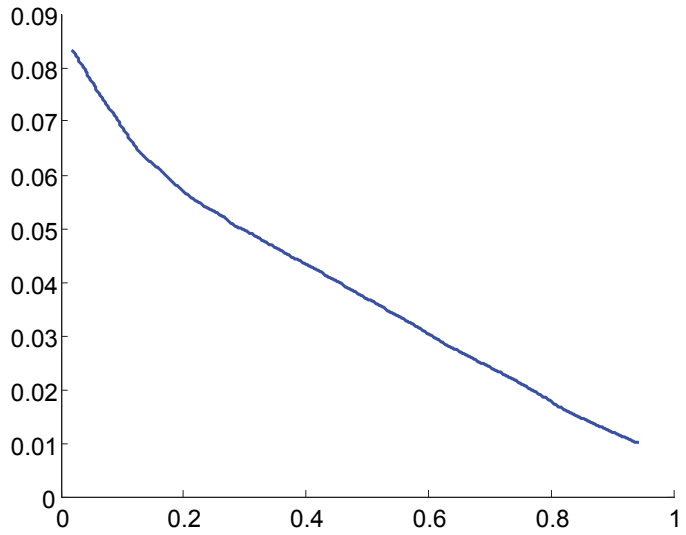

FIG. 5.3. *Robustness-loss curve: $L^*_{\epsilon_\ell}$ (vertical axis) vs. ϵ_ℓ (horizontal axis). ℓ is in the range* $8, \ldots, 6007$.

against ϵ_ℓ, represented on the horizontal axis, for ℓ values in the range $8, \dots, 6007$. The interpretation is that the value on the vertical axis is a guaranteed bound for $|s(t) - \hat{s}^*_{7277}(t)|$ with probability given by 1 minus the value on the horizontal axis. Based on Theorem 5.1, all the points in the robustness-loss curve are simultaneously guaranteed with confidence $1 - 6000 \cdot 10^{-10} = 1 - 6 \cdot 10^{-7}$.

6. Concluding remarks. In this paper random convex programs with L^1-regularization have been introduced. L^1-regularization allows one to shrink the number of optimization variables, and thereby enhance the generalization properties of the random convex program. Explicit formulas for evaluating the level of generalization have been derived.

In some applications, the scenarios $\delta^{(i)}$ are sampled from Δ by the user according to a probabilistic model. In other applications, the $\delta^{(i)}$'s come as data and the underlying probability $\mathbb{P}$ is not known (referring, e.g., to the minimax regression example of section 4.1, one can think of situations in which the samples $(t^{(i)}, s(t^{(i)}))$ are observed data). Importantly, the results of this paper are perfectly tailored to deal with this second setup as well. Indeed, knowledge of probability $\mathbb{P}$ is not needed to run the $\boldsymbol{L_1}$**–RCA** algorithm (since this algorithm uses only the scenarios), nor is knowledge of $\mathbb{P}$ required to apply the theoretical results (since all results in this paper are distribution-free, i.e., they hold irrespective of $\mathbb{P}$). This observation opens up important opportunities for applying the findings of this paper to signal processing problems and, more generally, to any data-based minimax optimization problem arising, e.g., in finance, classification, and engineering design.

The work presented in this paper refers to uncertain objective functions. Extending the results herein to uncertain constraints is certainly of interest.

Appendix A. MATLAB codes.

A.1. MATLAB code to solve (3.8) for N.

Function inputs: `eps` $= \epsilon$; `bet` $= \beta$; `p` $= p$; `d` $= d$; `q` $= q$.

REMARK. *In the function,* N *is computed by bisection;* N1 *is the initial lower bound, while* N2 *is the initial upper bound and corresponds to formula* (3.3).

Function findN

```
function N = findN(eps,bet,p,d,q)

N1 = q;
N2 = 2/eps*(log(1/bet) + q + (p-d+q)*log((p*exp(1))/(p-d+q)));

while N2-N1>1

        N = floor((N1+N2)/2);
        if (1/((p-d+q)*beta(d-q+1,p-d+q)))*(betainc(1-eps,N-q,q+1))>bet
        N1=N;
        else
        N2=N;
        end

end

N = N2
```

A.2. MATLAB code to compute ϵ_ℓ.

Function inputs: `l` $= \ell$; `N` $= N$; `bet` $= \beta$; `p` $= p$; `d` $= d$; `q` $= q$.

Function findepsl

```
function epsl = findepsl(l,N,bet,p,d,q)

eps1 = l/N;
eps2 = 1;

while eps2-eps1 > 1e-10
        epsl = (eps1+eps2)/2;
        if (1/((p-d+q)*beta(d-q+1,p-d+q)))*(betainc(1-epsl,N-l,l+1))>bet
        eps1 = epsl;
        else
        eps2 = epsl;
        end
end

epsl = eps2
```

Acknowledgment. We feel indebted with three anonymous reviewers for many insightful comments that helped to improve this manuscript.

REFERENCES

[1] T. Alamo, R. Tempo, and E. F. Camacho, *Randomized strategies for probabilistic solutions of uncertain feasibility and optimization problems*, IEEE Trans. Automat. Control, 54 (2009), pp. 2545–2559.

[2] A. R. Barron, *Universal approximation bounds for superpositions of a sigmoidal function*, IEEE Trans. Inform. Theory, 39 (1993), pp. 930–944.

[3] A. Ben-Tal and A. Nemirovski, *Robust convex optimization*, Math. Oper. Res., 23 (1998), pp. 769–805.

[4] A. Ben-Tal and A. Nemirovski, *On tractable approximations of uncertain linear matrix inequalities affected by interval uncertainty*, SIAM J. Optim., 12 (2002), pp. 811–833.

[5] G. Calafiore and M. C. Campi, *Uncertain convex programs: Randomized solutions and confidence levels*, Math. Program., 102 (2005), pp. 25–46.

[6] G. Calafiore and M. C. Campi, *The scenario approach to robust control design*, IEEE Trans. Automat. Control, 51 (2006), pp. 742–753.

[7] M. C. Campi and S. Garatti, *The exact feasibility of randomized solutions of uncertain convex programs*, SIAM J. Optim., 19 (2008), pp. 1211–1230.

[8] E. J. Candes, J. Romberg, and T. Tao, *Stable signal recovery from incomplete and inaccurate measurements*, Comm. Pure Appl. Math., 59 (2006), pp. 1207–1223.

[9] E. J. Candes and T. Tao, *Near optimal signal recovery from random projections: Universal encoding strategies?*, IEEE Trans. Inform. Theory, 52 (2006), pp. 5406–5425.

[10] E. J. Candes and T. Tao, *The Dantzig selector: Statistical estimation when p is much larger than n*, Ann. Statist., 35 (2007), pp. 2313–2351.

[11] E. J. Candes and M. B. Wakin, *An introduction to compressive sampling*, IEEE Signal Process. Mag., 21 (2008), pp. 21–30.

[12] C. Caramanis, S. Mannor, and H. Xu, *Robust optimization in machine learning*, in Optimization for Machine Learning, S. Sra, S. Nowozin, and S. Wright, eds., MIT Press, Cambridge, MA, 2011.

[13] S. S. Chen, D. L. Donoho, and M. A. Saunders, *Atomic decomposition by basis pursuit*, SIAM J. Sci. Comput., 20 (1998), pp. 33–61.

[14] H. Chernoff, *A measure of asymptotic efficiency for test of hypothesis based on the sum of observations*, Ann. Math. Statist., 23 (1952), pp. 493–507.

[15] D. Donoho, *Compressed sensing*, IEEE Trans. Inform. Theory, 52 (2006), pp. 1289–1306.

[16] D. Donoho, *For most large underdetermined systems of equations, the minimal ℓ^1-norm is the sparsest solution*, Comm. Pure Appl. Math., 59 (2006), pp. 797–829.
[17] D. Donoho and X. Huo, *Uncertainty principles and ideal atomic decomposition*, IEEE Trans. Inform. Theory, 47 (2001), pp. 2845–2862.
[18] E. Erdoğan and G. Iyengar, *Ambiguous chance constrained problems and robust optimization*, Math. Program. Ser. B, 107 (2006), pp. 37–61.
[19] G. Furnival and R. Wilson, *Regression by leaps and bounds*, Technometrics, 16 (1974), pp. 499–511.
[20] S. Garatti and M. C. Campi, *Modulating robustness in control design: Principles and algorithms*, IEEE Control Syst. Mag., 33 (2013), pp. 36–51.
[21] M. Grant and S. Boyd, *CVX: MATLAB Software for Disciplined Convex Programming*, http://cvxr.com/cvx/.
[22] M. Grant and S. Boyd, *Graph implementations for nonsmooth convex programs*, in Recent Advances in Learning and Control (a tribute to M. Vidyasagar), V. Blondel, S. Boyd, and H. Kimura, eds., Springer, New York, 2008, pp. 95–110.
[23] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed., Springer, New York, 2009.
[24] L. K. Jones, *A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training*, Ann. Statist., 20 (1992), pp. 608–613.
[25] K. Knight and W. Fu, *Asymptotics for lasso-type estimators*, Ann. Statist., 28 (2000), pp. 1356–1378.
[26] Y. Lin and H. Zhang, *Component selection and smoothing in smoothing spline analysis of variance models*, Ann. Statist., 34 (2006), pp. 2272–2297.
[27] J. Löfberg, *YALMIP: A toolbox for modeling and optimization in MATLAB*, in Proceedings of the CACSD Conference, Taipei, Taiwan, 2004.
[28] J. Luedtke and S. Ahmed, *A sample approximation approach for optimization with probabilistic constraints*, SIAM J. Optim., 19 (2008), pp. 674–699.
[29] N. Meinshausen, *Relaxed lasso*, Comput. Statist. Data Anal., 52 (2007), pp. 374–393.
[30] A. Nemirovski, *Several NP-hard problems arising in robust stability analysis*, Math. Control Signals Systems, 6 (1993), pp. 99–105.
[31] A. Nemirovski and A. Shapiro, *Convex approximations of chance constrained programs*, SIAM J. Optim., 17 (2006), pp. 969–996.
[32] H. Ohlsson, L. Ljung, and S. Boyd, *Segmentation of ARX-models using sum-of-norms regularization*, Automatica J. IFAC, 46 (2010), pp. 1107–1111.
[33] N. Ozay, M. Sznaier, C. Lagoa, and O. Camps, *A sparsification approach to set membership identification of a class of affine hybrid systems*, in Proceedings of the IEEE Conference on Decision and Control, Cancun, Mexico, 2008, pp. 123–130.
[34] B. K. Pagnoncelli, S. Ahmed, and A. Shapiro, *Sample average approximation method for chance constrained programming: Theory and applications*, J. Optim. Theory Appl., 142 (2009), pp. 399–416.
[35] S. Rosset and J. Zhu, *Piecewise linear regularized solution paths*, Ann. Statist., 35 (2007), pp. 1012–1030.
[36] A. N. Shiryaev, *Probability*, Springer, New York, 1996.
[37] R. Tibshirani, *Regression shrinkage and selection via the lasso*, J. Roy. Statist. Soc. Ser. B, 58 (1996), pp. 267–288.
[38] J. Tropp, *Greed is good: Algorithmic results for sparse approximation*, IEEE Trans. Inform. Theory, 50 (2004), pp. 2231–2242.
[39] V. Vapnik, *Statistical Learning Theory*, John Wiley, New York, 1996.
[40] M. Vidyasagar, *Statistical learning theory and randomized algorithms for control*, IEEE Control Syst. Mag., 18 (1998), pp. 69–85.
[41] H. Xu, C. Caramanis, and S. Mannor, *Robustness and regularization of support vector machines*, J. Mach. Learn. Res., 10 (2009), pp. 1485–1510.
[42] H. Xu, C. Caramanis, and S. Mannor, *Robust regression and Lasso*, IEEE Trans. Inform. Theory, 56 (2010), pp. 3561–3574.
[43] H. Xu, C. Caramanis, S. Mannor, and S. Yun, *Risk sensitive robust support vector machines*, in Proceedings of the IEEE Conference on Decision and Control, Shanghai, China, 2009.
[44] H. Xu and S. Mannor, *Robustness and generalization*, Machine Learning, 86 (2012), pp. 391–423.