



# Probabilistic iterative hard thresholding for sparse learning

Matteo Bergamaschi<sup>1</sup> · Andrea Cristofari<sup>2</sup> · Vyacheslav Kungurtsev<sup>3</sup> ·  
Francesco Rinaldi<sup>1</sup> 

Received: 24 October 2024 / Accepted: 2 August 2025 / Published online: 21 August 2025  
© The Author(s) 2025

## Abstract

For statistical modeling wherein the data regime is unfavorable in terms of dimensionality relative to the sample size, finding hidden sparsity in the relationship structure between variables can be critical in formulating an accurate statistical model. The so-called “ $\ell_0$  norm”, which counts the number of non-zero components in a vector, is a strong reliable mechanism of enforcing sparsity when incorporated into an optimization problem for minimizing the fit of a given model to a set of observations. However, in big data settings wherein noisy estimates of the gradient must be evaluated out of computational necessity, the literature is scant on methods that reliably converge. In this paper, we present an approach towards solving expectation objective optimization problems with cardinality constraints. We prove convergence of the underlying stochastic process and demonstrate the performance on two Machine Learning problems.

**Keywords** Cardinality constraint · Stochastic optimization

---

✉ Francesco Rinaldi  
rinaldi@math.unipd.it

Matteo Bergamaschi  
bergamas@math.unipd.it

Andrea Cristofari  
andrea.cristofari@uniroma2.it

Vyacheslav Kungurtsev  
kunguvya@fel.cvut.cz

- <sup>1</sup> Department of Mathematics “Tullio Levi-Civita”, University of Padua, Padova, Italy
- <sup>2</sup> Department of Civil Engineering and Computer Science Engineering, University of Rome “Tor Vergata”, Roma, Italy
- <sup>3</sup> Department of Civil Engineering and Computer Science Engineering, Czech Technical University in Prague, Kladno, Czechia

## 1 Introduction

In this paper, we consider the optimization problem defined as the minimization of an expectation objective subject to a constraint on the cardinality, that is, the number of non-zeros components in the decision vector. Formally,

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) := \mathbb{E}[F(x, \xi)] \\ \text{s.t.} \quad & \|x\|_0 \leq K, \end{aligned} \quad (1)$$

where  $f(\cdot)$  is  $L(f)$ -Lipschitz continuously differentiable, that is,  $\|\nabla f(x) - \nabla f(y)\| \leq L(f)\|x - y\|$  for all  $x, y \in \mathbb{R}^n$ , with  $L(f) > 0$ . We say that  $x \in C_K \subseteq \mathbb{R}^n$  if  $\|x\|_0 \leq K$  and thus a feasible  $x$  corresponds to  $x \in C_K$ .

This optimization problem is particularly important in data science applications. In particular, the expectation objective serves to quantify the minimization of some empirical loss function that enforces the fit of a statistical model to empirical data. Cardinality constraints enforce sparsity in the model, enabling the discovery of the most salient features as far as predicting the label.

Cardinality constraints present a significant challenge to optimization solvers. The so-called (as it is not, formally) zero “norm” is a discontinuous function that results in a highly non-convex and disconnected feasible set, as well as an unusual topology of stationary points and minimizers [1, 2]. Algorithmic development has been, as similar to many such problems, a parallel endeavor from the mathematical optimization and the machine learning communities. When dealing with a deterministic objective function, procedures attuned to the structure of the problem and seeking stationary points of various strength are presented, for instance, in [3]. Methods for deterministic optimization problems with sparse symmetric sets are proposed in, e.g., [4, 5], while methods for deterministic optimization problems with both cardinality and nonlinear constraints are described in, e.g., [6–12]. Simultaneously works appearing in machine learning conferences, e.g., [13–16], exhibit weak theoretical convergence guarantees, but appear to scale more adequately as far as numerical experience. Thus, an algorithm that enjoys both reliable performance together with strong theoretical guarantees, as sought for the high dimensional high data volume model fitting problems in contemporary data science, is as of yet unavailable.

In this paper we attempt to reconcile these two and present an algorithm that is associated with reasonably strong theoretical convergence guarantees, while at the same time able to solve large scale problems of interest in statistics and machine learning. To this end, we present a procedure under the framework of *Probabilistic Models*, which can be understood as a sequential linear Sample Average Approximation (SAA) scheme for solving problems with statistics in the objective function. First introduced in [17], then rediscovered with extensive analysis in [18, 19], this approach can exhibit asymptotic (and even worst case complexity) results to a local minimizer of the original problem, while still allowing the use of Newton-type second order iterations of subproblem solutions, and thus faster convergence as far as iteration count. The use of probabilistically accurate estimates within a certain bound in these methods permit a rather flexible approach to estimating the gradient, including techniques that introduce bias, while foregoing the necessity of a stepsize

asymptotically diminishing to zero. However, asymptotic accurate convergence still requires increasing the batch size, so the tradeoffs in precision and certainty relative to computation become apparent, and adaptive for the user, in deciding at which point to stop the algorithm and return the current iterate as an estimate of the solution.

As contemporary Machine Learning applications, we shall consider Adversarial Attacks (see, e.g., [20–22] and references therein for further details) and Probabilistic Graphical Model training (see, e.g., [23, 24] and references therein for further details). In this paper, we shall see how the use of a stochastic gradient and hard sparsity constraint can improve the performance and model quality in the considered problems.

The paper is organized as follows: In Sect. 2, we introduce some basic definitions and preliminary results related to optimality conditions of problem (1) that ease the theoretical analysis. We then describe the details of the proposed algorithmic scheme in Sect. 3. We then prove almost sure convergence to suitable stationary points in Sect. 4. Numerical results on some relevant Machine Learning applications are reported Sect. 5. Finally, we draw some conclusions and discuss some possible extensions in Sect. 6.

## 2 Background

Cardinality constrained optimization presents an extensive hierarchy of stationarity conditions, as due to the geometric complexity of the feasible set. This necessitates specialized notions of projection and presents complications due to the projection operation’s generic non-uniqueness.

**Definitions and Preliminaries** Given a vector  $x \in \mathbb{R}^n$ , we denote the  $i$ th component of  $x$  by  $[x]_i$  and the subvector related to the components with indices in  $I \subseteq [1 : n]$  by  $[x]_I$ , while the active and inactive set of  $x$  are respectively denoted by

$$I_A(x) := \{i \in \{1, \dots, n\}, [x]_i = 0\}, \quad I_I(x) := \{i \in \{1, \dots, n\}, [x]_i \neq 0\}.$$

A set  $T \subseteq \{1, \dots, n\}$  is a *super-support* of  $x \in C_K$  if  $I_I(x) \subseteq T$  and  $|T| = K$ . Let the permutation group of  $\{1, \dots, n\}$  be denoted as  $\Sigma_n$  and, for a permutation  $\sigma \in \Sigma_n$ , we write  $[x^\sigma]_i = x_{\sigma(i)}$ , where  $\sigma(i)$  denotes the  $i$ th element of  $\sigma$ . For a vector  $x \in \mathbb{R}^n$  we denote with  $M_i(x)$  the  $i$ -th largest absolute-value component of  $x$ , thus we have  $M_1(x) \geq M_2(x) \geq \dots \geq M_n(x)$ . Let  $\sigma^O$  correspond to sorting by this ordering.

We finally define the orthogonal projection as

$$P_{C_K}(x) \in \arg \min\{\|z - x\|^2, z \in C_K\} = \{z \in C_K : [z]_{\sigma^O(i)} = M_i(x), i \leq K, [z]_{\sigma^O(i)} = 0, i > K\},$$

that is, an  $n$ -length vector consisting of the  $K$  components of  $x$  with the largest absolute value. Such operator, as already highlighted in the previous section, is not single-valued due to the inherent non-convexity of the set  $C_K$ . For instance consider projecting  $(3, 1, 1)^T$  onto  $C_2$ . This gives two different points, that is  $P_{C_2}((3, 1, 1)^T) = \{(3, 1, 0)^T, (3, 0, 1)^T\}$ . A theoretical tool of this kind plays a

critical role in the development of algorithms for sparsity-constrained optimization (see, e.g., [3, Sect. 2] for a discussion on this matter).

### Optimality Conditions

Now we define several optimality conditions for (1), borrowing heavily from [3]. Observe that a notable characteristic of cardinality constrained optimization is the presence of a hierarchy of optimality conditions, that is, a number of conditions that hold at optimal points that range across levels of restrictiveness or specificity.

When restricted to a specific support, the “no descent directions” rule still provides a necessary optimality condition, which is referred to as basic feasibility. For a full support, that is for  $\|x^*\|_0 = K$ , this condition aligns with standard stationarity conditions of non-negative directional derivative along any direction in the linearization of the feasible tangent cone. This linearization only includes directions with nonzero components restricted to  $I_{\mathcal{I}}(x^*)$ . If the support is not full, i.e.,  $\|x^*\|_0 < K$ , the stationarity condition must hold for any potential super support set of  $x^*$ . The specific definition is given below:

**Definition 1**  $x^* \in C_K$  is *Basic Feasible* (BF) for problem (1) when

1.  $\nabla f(x^*) = 0$ , if  $\|x^*\|_0 < K$ ,
2.  $[\nabla f(x^*)]_i = 0$  for all  $i \in I_{\mathcal{I}}(x^*)$ , if  $\|x^*\|_0 = K$ .

We thus have that when a point  $x^* \in C_K$  is optimal for problem (1), then  $x^*$  is a BF point (see Theorem 1 in [3]). The BF property is however a relatively weak necessary condition for optimality, meaning that a problem could have a potentially large set of suboptimal BF points. Consequently, stronger necessary conditions are required to achieve higher quality solutions. This is why we use  $L$ -stationarity, a stationarity concept with notation similar to minimization over a simple convex set which, in the case of cardinality constraints, considers the ranking of the absolute value of the gradient components.

**Definition 2** Given  $L > 0$ , we say that  $x^* \in C_K$  is  *$L$ -stationary* for problem (1) when

$$x^* \in P_{C_K} \left( x^* - \frac{1}{L} \nabla f(x^*) \right). \quad (2)$$

An equivalent analytic property of  $L$ -stationarity is given by the following lemma.

**Lemma 1** [3, Lemma 2.2]  $L$ -stationarity at  $x^*$  is equivalent to  $\|x^*\|_0 \leq K$  and

$$|[\nabla f(x^*)]_i| \begin{cases} \leq LM_K(x^*) & i \in I_{\mathcal{A}}(x^*) \\ = 0 & i \in I_{\mathcal{I}}(x^*). \end{cases}$$

It can be shown that  $L$ -stationarity is a more restrictive condition than Basic Feasibility:

**Corollary 1** [3, Corollary 2.1] Suppose that  $x^* \in C_k$  is an L-stationary of problem (1) for some  $L$ . Then  $x^*$  is BF for problem (1).

In addition, the likely intuition that the  $L$ -stationarity is related to the gradient Lipschitz constant is correct:

**Theorem 1** [3, Theorem 2.2] If  $x^*$  is an optimal solution for problem (1) then it is  $L$ -stationary for all  $L > L(f)$ .

To see the distinction between BF and L-stationary, we consider  $x^* = (0, 1)$  with  $\nabla f(x^*) = (-4, 0)$  and  $K = 1$ . It is easy to see that  $x^*$  satisfies BF but not  $L$ -stationarity, with  $L < 4$ . This example illustrates how the consideration of the structure of the feasible set for cardinality constrained optimization introduces the necessity of incorporating combinatorial properties associated with ranking gradient components.

In this context,  $L$ -stationarity is stronger than BF in the sense that the latter only considers a linearized feasible direction stationarity measure, which is also done in the work on sequential conditions in [25]. Observe that given  $x \in \mathbb{R}^n$  and any  $d \in \mathbb{R}^n$  such that  $d_i \neq 0$  for some  $i \in I_A(x)$ ,  $x + td \in C_K$  for small  $t > 0$  is only possible if  $\|x\|_0 < K$ . Thus, if  $\|x\|_0 = K$ , for there to be any such  $d_i$ , there would have to be some  $w \in \mathbb{R}^n$  such that  $[w]_j = -[x]_j$  for  $j \in I_X(x)$ , and the new point would have to be of the form  $x + w + td$ . In other words, any feasible direction from which a zero component becomes non-zero, would first require a swap, that is the assignment to zero, for another non-zero component, in order to guarantee the cardinality constraint to be satisfied. Thus there is no feasible direction in which a zero component becomes non-zero when the cardinality constraint is active.  $L$ -stationarity enables an exploration of gradient vector components in order to propose directions with distinct support from the current point, which is more aligned to the spirit of cardinality-constrained optimization problems.

**Iterative Hard Thresholding** An important algorithmic tool, often used in the machine learning literature to deal with cardinality-constrained problems, is the *Hard Thresholding Operator* (see, e.g., [3, 26] for further details). Consider the operator  $HT(v)$  applied to a vector  $v$  as one that projects  $v$  onto the sparsity constraint, i.e.,

$$HT(v) \in \arg \min_w \{\|v - w\|, \|w\|_0 \leq K\} := P_{C_K}(v). \tag{3}$$

Iterative Hard Thresholding (IHT) is hence an optimization algorithm designed to solve deterministic sparse optimization problems, particularly those involving cardinality constraints, which iteratively updates a solution by combining gradient steps with the Hard Thresholding operator:

$$x_{k+1} \in P_{C_K}(x_k - \alpha \nabla f(x_k)), \quad k = 0, 1, 2, \dots \tag{4}$$

This fixed point method is able to enforce the L-stationary condition in the accumulation points of the generated sequence when a suitable stepsize  $\alpha > 0$  is chosen (see, e.g., [3] and references therein for further details).

### 3 Probabilistic iterative hard thresholding algorithm

This section introduces the Probabilistic Iterative Hard Thresholding algorithm, a new method that enables to tackle cardinality-constrained stochastic optimization problems, and the core ideas related to the building blocks of the algorithm, that is the Pseudo Hard Thresholding Operator and the stochastic function estimates.

**Pseudo Hard Thresholding Operator** Given  $v \in \mathbb{R}^n$ , let us define  $\tilde{\Sigma}(v)$  as the set of all *sorting permutations*, that is,  $\sigma \in \tilde{\Sigma}(v)$  if and only if  $[v]_{\sigma(1)} \geq [v]_{\sigma(2)} \geq \dots \geq [v]_{\sigma(n)}$ , where non-uniqueness of the operation arises in case of ties. Recall that, the sparse projection operation  $P_{C_K}(v)$  for a vector  $v$  amounts to sorting  $\{|[v]_i|\}$  in order to define some  $\sigma \in \tilde{\Sigma}(|v|)$  and then keeping the  $K$  largest magnitude components of  $v$  while setting the rest to zero.

As observed above, an algorithmic iterative descent procedure would involve the negative of the gradient of  $f$ , or an estimate thereof. Indeed, as the objective function is an expectation, we do not have access to the exact value of the  $\nabla f(x)$  and hence the magnitude ranking of its components. Thus, it is necessary for us to use noisy gradient estimates  $\nabla F(x, \hat{\xi})$  to try to guess the actual ranking of the component magnitudes.

Asymptotically, we want to ensure that this sparse projector estimates the true ranking at any limit point. Given that the sequence of iterates provides incrementally and asymptotically increasing sample sizes of points in the neighborhood of the solution limit point, this presents a natural opportunity to use the algorithm iterate sequence itself to perform this estimate. By relying on consistency in the asymptotic sampling regime, we obtain statistical guarantees on accurate identification.

Let  $x_k$  correspond to the current iterate of the given algorithm. Now we define our particular sequential ranking estimate for the magnitude of the vector components related to the gradient of  $f(x_k)$ .

During the earlier iterations of the algorithm, there accumulated a set of permutations  $S_k = \{\sigma^{(j)}\}_{j \in [J]}$ ,  $\sigma^{(j)} \in \Sigma_n$  with coefficient weights  $\{\omega^{(j)}\}_{j \in [J]}$ ,  $\omega \in \Delta_J$ , where we use  $\Delta_m$  to denote the unit simplex of dimension  $m$ , and  $[J] = \{1, \dots, J\}$ , with  $J = |S_k|$ . Now we compute a new gradient estimate at  $k$ , and perform the inductive step on the procedure of updating the set  $S_k$ .

Specifically, given a noisy evaluation  $g_k \approx \nabla f(x_k)$ , we take a *clipped* gradient step, wherein we step in the negative direction of the scaled negative gradient estimate  $-\alpha \min \left\{ 1, \frac{\delta_k}{\alpha \|g_k\|} \right\} g_k$ , with  $\delta_k > 0$  the clipping level, which is updated during the iterations (see below), and  $\alpha > 0$  a constant which, roughly speaking, represents a sort of stepsize along the chosen direction. It is easy to see that the clipped gradient step projects the term  $-\alpha g_k$  within a ball of radius  $\delta_k$ . Then we consider a sorting permutation of the vector:

$$\sigma_k \in \tilde{\Sigma} \left( x_k - \alpha \min \left\{ 1, \frac{\delta_k}{\alpha \|g_k\|} \right\} g_k \right), \quad (5)$$

Then, taking

$$I_k = \{\sigma_k(i), i = 1, \dots, K\}, \tag{6}$$

that is, the set of indices whose components are largest according to  $\sigma_k$ , we introduce the **Pseudo Hard Thresholding** operator corresponding to iteration  $k$ , defined as follows:

$$\text{PHT}^k(v) \in \arg \min_w \{\|v - w\|, [w]_{[1:n] \setminus I_k} = 0\} = P_{I_k}(v), \tag{7}$$

where  $P_{I_k}(v)$  is the projection of  $v$  over the subspace of  $\mathbb{R}^n$  defined by those points having support in  $I_k$ . As highlighted above, we take a *clipped* gradient step and then apply our Pseudo-Hard Thresholding operator on this point:

$$\hat{x}_k = \text{PHT}^k \left( x_k - \alpha \min \left\{ 1, \frac{\delta_k}{\alpha \|g_k\|} \right\} g_k \right) = P_{I_k} \left( x_k - \alpha \min \left\{ 1, \frac{\delta_k}{\alpha \|g_k\|} \right\} g_k \right). \tag{8}$$

Observe that unlike Hard Thresholding, the Pseudo Hard Thresholding operator can be computed in a straightforward closed form expression of simply setting the components not corresponding to the estimated top  $K$  to be zero, that is,

$$[\hat{x}_k]_i = \begin{cases} 0 & i \notin I_k, \\ \left[ x_k - \alpha \min \left\{ 1, \frac{\delta_k}{\alpha \|g_k\|} \right\} g_k \right]_i & i \in I_k. \end{cases} \tag{9}$$

Note that, using known properties of the projection operator [27], for all  $v \in \mathbb{R}^n$  we can write  $(x_k - v - P_{I_k}(x_k - v))^T(x_k - P_{I_k}(x_k - v)) \leq 0$ . So, recalling (8), we have

$$g_k^T(\hat{x}_k - x_k) \leq -\frac{1}{\alpha} \max \left\{ 1, \frac{\alpha \|g_k\|}{\delta_k} \right\} \|\hat{x}_k - x_k\|^2.$$

This presents an opportunity to establish a descent lemma adapted to the context of cardinality constrained optimization problems. To this end, define

$$h_k(y) = f(x_k) + g_k^T(y - x_k), \tag{10}$$

so that

$$h_k(\hat{x}_k) - h_k(x_k) = g_k^T(\hat{x}_k - x_k) \leq -\frac{1}{\alpha} \max \left\{ 1, \frac{\alpha \|g_k\|}{\delta_k} \right\} \|\hat{x}_k - x_k\|^2 \leq -\frac{1}{\alpha} \|\hat{x}_k - x_k\|^2. \tag{11}$$

**Accuracy Estimates** In our algorithm we require that both the function values  $f(x_k)$  and  $f(x_k + s_k)$  as well as the first-order models used to compute the step are sufficiently accurate with high probability. Here we present some standard definitions that make these notions precise. In particular, the following definitions of accurate function estimates and of accurate models are similar to the ones in [19].

**Definition 3** Define  $s_k = \hat{x}_k - x_k$ . The function estimates  $f_k^0$  and  $f_k^s$  are  $\varepsilon_f$ -accurate estimates of  $f(x_k)$  and  $f(x_k + s_k)$ , respectively, for a given  $\delta_k$  if

$$|f_k^0 - f(x_k)| \leq \varepsilon_f \delta_k^2 \quad \text{and} \quad |f_k^s - f(x_k + s_k)| \leq \varepsilon_f \delta_k^2. \quad (12)$$

**Definition 4** The model for generating the iterate is  $\kappa$ - $\delta_k$ , or  $(\kappa_f, \kappa_g)$ - $\delta_k$  accurate when

$$\|\nabla f(y) - g_k\| \leq \kappa_g \delta_k \quad \text{and} \quad |f(y) - f(x_k) - g_k^T(y - x_k)| \leq \kappa_f \|y - x_k\| \delta_k^2 \quad (13)$$

for all  $y$  such that  $[y]_{I_k} \in B([x_k]_{I_k}, \delta_k)$ .

In the following proposition we report a result implied by equation (13) that will be useful for the convergence analysis.

**Proposition 2** If (13) holds for all  $y$  such that  $[y]_{I_k} \in B([x_k]_{I_k}, \delta_k)$ , then

$$\|\nabla f(y) - g_k\|_{I_k} \leq \kappa_g \delta_k \quad \text{and} \quad |f(y) - f(x_k) - [g_k]_{I_k}^T [y - x_k]_{I_k}| \leq \kappa_f \| [y - x_k]_{I_k} \| \delta_k^2 \quad (14)$$

for all  $y$  such that  $[y]_{I_k} \in B([x_k]_{I_k}, \delta_k)$ .

**Proof** Let (13) hold. It is straightforward to verify that the first inequality in (14) follows from the first inequality in (13) since  $\|\nabla f(y) - g_k\|_{I_k} \leq \|\nabla f(y) - g_k\|$ . To show that the second inequality in (14) holds as well, assume by contradiction that there exists  $y \in \mathbb{R}^n$  such that  $[y]_{I_k} \in B([x_k]_{I_k}, \delta_k)$  and

$$|f(y) - f(x_k) - [g_k]_{I_k}^T [y - x_k]_{I_k}| > \kappa_f \| [y - x_k]_{I_k} \| \delta_k^2.$$

Now, define  $\tilde{y} \in \mathbb{R}^n$  as follows:

$$[\tilde{y}]_i = \begin{cases} [y]_i & \text{if } i \in I_k, \\ [x_k]_i & \text{if } i \notin I_k. \end{cases}$$

Then,  $[\tilde{y}]_{I_k} \in B([x_k]_{I_k}, \delta_k)$  and

$$|f(y) - f(x_k) - g_k^T(\tilde{y} - x_k)| = |f(y) - f(x_k) - [g_k]_{I_k}^T [\tilde{y} - x_k]_{I_k}| > \kappa_f \| [y - x_k]_{I_k} \| \delta_k^2 = \kappa_f \| y - x_k \| \delta_k^2,$$

thus contradicting the second inequality in (13).  $\square$

**Algorithm Description** See Algorithm 1 for a summary of the scheme we present in this paper. At the initialization, a feasible starting point  $x_0 \in C_K$  and a step  $\delta_0$  are chosen. Then, at each iteration, a random variable  $\xi_k$  is sampled from a distribution  $\Xi$  and the gradient estimate  $g_k = \nabla F(x_k, \xi_k)$  at the point  $x_k$  with respect to  $x$  with the realization defined by sample (e.g., minibatch)  $\xi_k$  is calculated (see Step 3 of the algorithm). In Step 4, the sorting permutation  $\sigma_k$  with largest weight is selected and used to define the set of indices  $I_k$  related to the largest components.

The Pseudo Hard Thresholding operator is then used to calculate the trial point  $\hat{x}_k$ . The algorithm hence generates an estimate of the true objective function  $f$  at the trial point  $\hat{x}_k$ . By also computing an estimate of  $f$  at the current point  $x_k$  in Step 7, it can perform a test, in Step 8, on whether there is a sufficient reduction of the model. If so, we have a *successful iteration*, the iterate is updated to equal the computed estimate  $\hat{x}_k$ . Moreover, the parameter  $\delta_k$  used to define the required accuracy conditions on the gradient estimate as well as the clipping level associated with the step normalization is increased, as long as it is smaller than some large threshold  $\delta_{max}$ . This relaxation of the subproblem requirements in stringency of accuracy and descent permits for potentially larger steps at the next iteration, promoting faster convergence when estimates are an accurate representation of the underlying objective function (see Step 9).

Otherwise, the sample estimate for sufficient decrease is not satisfied, we have an *unsuccessful iteration*, the  $\delta_k$  is reduced, and the iterate stays the same (see Step 11). However, it is important to note that a successful iteration does not necessarily give a reduction of the true function  $f$ . In fact, such a function is not available and the acceptance condition is based only on estimates of  $f(x_k)$  and  $f(\hat{x}_k)$ .

---

```

1: Initialization:  $x_0 \in C_K, \delta_0 \in (0, \delta_{max}]$ , Parameters  $\delta_{max} > 0, \gamma \in (0, 1)$ 
2: for  $k = 0, 1, 2, \dots$  do
3:   Sample a minibatch  $\xi_k \sim \Xi$  and compute  $g_k = \nabla F(x_k, \xi_k)$ 
4:   Compute  $\sigma_k$  by (5)
5:   Compute  $\hat{x}_k$  from the Pseudo Hard Thresholding (9)
6:   Compute stochastic estimates  $f_k^s \approx f(\hat{x}_k), f_k^0 \approx f(x_k)$ 
7:   if  $\frac{f_k^0 - f_k^s}{\|g_k\|_{I_k} \delta_k} \geq \eta_1$  and  $\|g_k\|_{I_k} \geq \eta_2 \delta_k$  then
8:     Set  $\delta_{k+1} = \min\{\gamma \delta_k, \delta_{max}\}$ , let  $x_{k+1} = \hat{x}_k$ 
9:   else
10:    Set  $\delta_{k+1} = \gamma^{-1} \delta_k$ , let  $x_{k+1} = x_k$ 
11:   end if
12: end for

```

---

**Algorithm 1** Probabilistic Iterative Hard Thresholding

### 4 Convergence theory

Now we develop our analysis for justifying the long term convergence of Algorithm 1 based on classic arguments on probabilistic models given in [19](see also [18, 28]). To this end, we remark that the iterates, being dependent on random function and gradient estimates, define a stochastic process  $X_k$ . The algorithm itself is a realization, thus denoting  $x_k = X_k(\omega), \delta_k = \Delta_k(\omega)$ , etc. for  $\omega$  the random element defining the realization. Similar as to the previous works, we consider a filtration with the sigma algebra  $\mathcal{F}_k$  defining the start of the iteration and  $\mathcal{F}_{k+\frac{1}{2}}$  defining the algebra after the minibatch has been sampled and  $g_k$  computed. This filtration will be implicit in the statements of the convergence results.

We begin with a standard assumption that gives probability bounds on the accuracy of the linear model defined by  $\{G_k\}$ , with  $G_k(\omega) = \nabla F(X_k, \omega)$  and the esti-

mates  $\{F_k^0, F_k^s\}$  defined similarly. To this end define  $\theta, \beta$  to be the probabilities that  $\{G_k\}$  is  $\kappa\text{-}\delta_k$ -accurate, and  $\{F_k^0, F_k^s\}$  are  $\varepsilon_f$ -accurate, respectively.

**Assumption 1** Given  $\theta, \beta \in (0, 1)$  and  $\varepsilon_f > 0$ , there exist  $\kappa_g, \kappa_f$  such that the sequence of gradient estimates  $\{G_k\}$  and function estimates  $\{F_k^0, F_k^s\}$  generated by Algorithm 1 are, respectively,  $\kappa\text{-}\delta_k$ -accurate with probability  $\theta$  as per Definition 4, and  $\varepsilon_f$ -accurate with probability  $\beta$  as per Definition 3.

One can expect that when the estimates are sufficiently close to their deterministic counterparts, the classical sparse optimization theory, namely [3, Lemma 3.1] provides for the guarantee of function decrease. Indeed, one can derive the following lemma which also functionally corresponds to [19, Lemma 4.5].

**Lemma 2** If the model for generating the iterate  $k$  is  $\kappa\text{-}\delta_k$  accurate according to Definition 4, with  $\hat{x}_k$  and  $\delta_k$  being such that

$$\delta_k \leq \frac{1}{2\alpha\kappa_f\delta_{max}} \|x_k - \hat{x}_k\|, \quad (15)$$

then

$$f(x_k) - f(\hat{x}_k) \geq \frac{1}{2\alpha} \|\hat{x}_k - x_k\|^2. \quad (16)$$

**Proof** Using the definition of  $h_k$  given in (10), we can write

$$\begin{aligned} f(\hat{x}_k) - f(x_k) &= f(\hat{x}_k) - h_k(\hat{x}_k) + h_k(\hat{x}_k) - h_k(x_k) + h_k(x_k) - f(x_k) \\ &= f(\hat{x}_k) - h_k(\hat{x}_k) + h_k(\hat{x}_k) - h_k(x_k) \\ &= f(\hat{x}_k) - f(x_k) - g_k^T(\hat{x}_k - x_k) + g_k^T(\hat{x}_k - x_k) \\ &\leq \kappa_f \|x_k - \hat{x}_k\| \delta_k^2 + g_k^T(\hat{x}_k - x_k), \end{aligned}$$

where the inequality follows from the second condition in (13). Using (11), we also have that

$$g_k^T(\hat{x}_k - x_k) \leq -\frac{1}{\alpha} \|\hat{x}_k - x_k\|^2.$$

Then, we obtain

$$\begin{aligned} f(\hat{x}_k) - f(x_k) &\leq \kappa_f \|x_k - \hat{x}_k\| \delta_k^2 - \frac{1}{\alpha} \|\hat{x}_k - x_k\|^2 \\ &\leq \kappa_f \delta_{max} \|x_k - \hat{x}_k\| \delta_k - \frac{1}{\alpha} \|\hat{x}_k - x_k\|^2, \end{aligned} \quad (17)$$

where the last inequality follows from the fact that  $\delta_k \leq \delta_{max}$ . Moreover, (15) implies that

$$\kappa_f \delta_{max} \|x_k - \hat{x}_k\| \delta_k \leq \frac{1}{2\alpha} \|\hat{x}_k - x_k\|^2.$$

Using this inequality in (17), the desired result follows. □

Now, taking inspiration from [19, Lemma 4.6], we can bound the decrease with respect to the projected real gradient.

**Lemma 3** If the model for generating the iterate  $k$  is  $\kappa\text{-}\delta_k$  accurate according to Definition 4 and

$$\delta_k \leq a \left\| x_k - P_{I_k} \left( x_k - \alpha \min \left\{ 1, \frac{\delta_k}{\alpha \|\nabla f(x_k)\|} \right\} \nabla f(x_k) \right) \right\|, \tag{18}$$

where

$$a = \frac{1}{2\alpha\kappa_f\delta_{max} + 2\sqrt{K}} \tag{19}$$

and

$$\alpha > \frac{\sqrt{K}}{\kappa_f\delta_{max}}, \tag{20}$$

then

$$f(x_k) - f(\hat{x}_k) \geq c \left\| x_k - P_{I_k} \left( x_k - \alpha \min \left\{ 1, \frac{\delta_k}{\alpha \|\nabla f(x_k)\|} \right\} \nabla f(x_k) \right) \right\|^2, \tag{21}$$

with

$$c = \frac{1 - 4a\sqrt{K}}{2\alpha} > 0.$$

**Proof** We can write

$$\begin{aligned} & \left\| x_k - P_{I_k} \left( x_k - \alpha \min \left\{ 1, \frac{\delta_k}{\alpha \|\nabla f(x_k)\|} \right\} \nabla f(x_k) \right) \right\| \leq \\ & \|x_k - \hat{x}_k\| + \left\| \hat{x}_k - P_{I_k} \left( x_k - \alpha \min \left\{ 1, \frac{\delta_k}{\alpha \|\nabla f(x_k)\|} \right\} \nabla f(x_k) \right) \right\|. \end{aligned}$$

Using (8), we get

$$\begin{aligned} & \left\| x_k - P_{I_k} \left( x_k - \alpha \min \left\{ 1, \frac{\delta_k}{\alpha \|\nabla f(x_k)\|} \right\} \nabla f(x_k) \right) \right\| = \\ & \|x_k - \hat{x}_k\| + \left\| \alpha \min \left\{ 1, \frac{\delta_k}{\alpha \|g_k\|} \right\} [g_k]_{I_k} - \alpha \min \left\{ 1, \frac{\delta_k}{\alpha \|\nabla f(x_k)\|} \right\} [\nabla f(x_k)]_{I_k} \right\| = \quad (22) \\ & \|x_k - \hat{x}_k\| + \delta_k \left\| \min \left\{ \frac{\alpha \|g_k\|}{\delta_k}, 1 \right\} \frac{[g_k]_{I_k}}{\|g_k\|} - \min \left\{ \frac{\alpha \|\nabla f(x_k)\|}{\delta_k}, 1 \right\} \frac{[\nabla f(x_k)]_{I_k}}{\|\nabla f(x_k)\|} \right\| \leq \\ & \|x_k - \hat{x}_k\| + 2\sqrt{K}\delta_k, \end{aligned}$$

where the last inequality follows from the fact that  $\|u - v\| \leq \sqrt{K}\|u - v\|_\infty \leq 2\sqrt{K}$  for all  $u, v \in \mathbb{R}^K$  such that  $\|u\| \leq 1$  and  $\|v\| \leq 1$ . From (18), the first term in (22) is greater or equal than  $\delta_k/a$ , leading to

$$\frac{\delta_k}{a} \leq \|x_k - \hat{x}_k\| + 2\sqrt{K}\delta_k.$$

Using the definition of  $a$  given in (19), it follows that (15) is satisfied and we can apply Lemma 2, obtaining

$$f(x_k) - f(\hat{x}_k) \geq \frac{1}{2\alpha} \|\hat{x}_k - x_k\|^2. \quad (23)$$

Finally, in order to lower bound the right-hand side term in the above inequality, using (22) we can write

$$\begin{aligned} \|x_k - \hat{x}_k\|^2 & \geq \left( \left\| x_k - P_{I_k} \left( x_k - \alpha \min \left\{ 1, \frac{\delta_k}{\alpha \|\nabla f(x_k)\|} \right\} \nabla f(x_k) \right) \right\| - 2\sqrt{K}\delta_k \right)^2 \\ & \geq \left\| x_k - P_{I_k} \left( x_k - \alpha \min \left\{ 1, \frac{\delta_k}{\alpha \|\nabla f(x_k)\|} \right\} \nabla f(x_k) \right) \right\|^2 + \\ & \quad - 4\sqrt{K}\delta_k \left\| x_k - P_{I_k} \left( x_k - \alpha \min \left\{ 1, \frac{\delta_k}{\alpha \|\nabla f(x_k)\|} \right\} \nabla f(x_k) \right) \right\| \\ & \geq (1 - 4a\sqrt{K}) \left\| x_k - P_{I_k} \left( x_k - \alpha \min \left\{ 1, \frac{\delta_k}{\alpha \|\nabla f(x_k)\|} \right\} \nabla f(x_k) \right) \right\|^2, \end{aligned}$$

where the last inequality follows from (18). From (20), it also follows that  $c > 0$ , thus leading to the desired result.  $\square$

The next lemma states conditions on  $\delta_k$  to guarantee that an iteration is successful, similarly as in [19, Lemma 4.7].

**Lemma 4** If, at iteration  $k$ , the estimates  $f_k^0, f_k^s$  are  $\varepsilon_f$ -accurate according to Definition 3 and the model is  $\kappa\text{-}\delta_k$  accurate according to Definition 4, with

$$\delta_k \leq \min \left\{ \frac{1}{\eta_2}, \frac{1 - \eta_1}{2\varepsilon_f + \kappa_f \delta_{max}} \right\} \|[g_k]_{I_k}\|,$$

then the step is accepted.

**Proof** Define

$$\rho_k = \frac{f_k^0 - f_k^s}{\|[g_k]_{I_k}\| \delta_k}.$$

Using (12) and (14), we can write

$$\begin{aligned} \rho_k &= \frac{f_k^0 - f(x_k)}{\|[g_k]_{I_k}\| \delta_k} + \frac{f(x_k) - f(\hat{x}_k)}{\|[g_k]_{I_k}\| \delta_k} + \frac{f(\hat{x}_k) - f_k^s}{\|[g_k]_{I_k}\| \delta_k} \\ &\leq \frac{2\varepsilon_f \delta_k}{\|[g_k]_{I_k}\|} + \frac{|[g_k]_{I_k}^T [\hat{x}_k - x_k]_{I_k}| + \kappa_f \|\hat{x}_k - x_k\|_{I_k} \delta_k^2}{\|[g_k]_{I_k}\| \delta_k} \\ &\leq \frac{2\varepsilon_f \delta_k}{\|[g_k]_{I_k}\|} + 1 + \frac{\kappa_f \delta_{max} \delta_k}{\|[g_k]_{I_k}\|}, \end{aligned}$$

where the last inequality follows from the fact that  $|[g_k]_{I_k}^T [\hat{x}_k - x_k]_{I_k}| \leq \|[g_k]_{I_k}\| \|x_k - \hat{x}_k\|_{I_k}$  and  $\|\hat{x}_k - x_k\|_{I_k} \leq \delta_k \leq \delta_{max}$ . Then

$$|\rho_k - 1| \leq \frac{(2\varepsilon_f + \kappa_f \delta_{max}) \delta_k}{\|[g_k]_{I_k}\|} \leq 1 - \eta_1,$$

where we have used the assumption on  $\delta_k$  in the last inequality. Hence,  $\rho_k \geq \eta_1$ . Since we have also assumed that  $\|[g_k]_{I_k}\| \geq \eta_2 \delta_k$ , from the instructions of the algorithm (see line 8 of Algorithm 1) it follows that the step is accepted.  $\square$

**Lemma 5** If the estimates  $f_k^0, f_k^s$  at iteration  $k$  are  $\varepsilon_f$ -accurate according to Definition 3 with  $\varepsilon_f < (\eta_1 \eta_2)/2$  and the step is accepted, then

$$f(x_{k+1}) - f(x_k) \leq -C \|\delta_k\|^2,$$

with  $C = \eta_1 \eta_2 - 2\varepsilon_f > 0$ .

**Proof** Since the step is accepted, from the instructions of the algorithm (see line 8 of Algorithm 1) we can write

$$f_k^0 - f_k^s \geq \eta_1 \| [g_k]_{I_k} \| \delta_k \geq \eta_1 \eta_2 \delta_k^2. \quad (24)$$

Moreover,

$$f(x_k + s_k) - f(x_k) = f(x_k + s_k) - f_k^s + f_k^s - f_k^0 + f_k^0 - f(x_k) \leq 2\epsilon_f \delta_k^2 - \eta_1 \eta_2 \delta_k^2,$$

where the inequality follows from (12) and (24). Then, using the definition of  $C$  given in the assertion, the desired result follows.  $\square$

Now we define the stochastic process

$$\Phi_k := \nu f(x_k) + (1 - \nu) \delta_k^2. \quad (25)$$

The next theorem is along the lines of Theorem 4.11 in [19]. The result requires a compactness assumption, which we present first.

**Assumption 2** Given  $\delta_{\max}$  and some initial guess  $x_0$ , let  $\mathcal{L}(x_0, \delta_{\max})$  be the set containing all the iterates generated by the algorithm, noting that this depends on the stochastic realization of the iterates and gradient estimates. Furthermore, let

$$\bar{\mathcal{L}}(x_0, \delta_{\max}) := \bigcup_{x \in \mathcal{L}(x_0, \delta_{\max})} B(x, \delta_{\max})$$

be the union of  $\delta_{\max}$  radius balls around all of the iterates.

Assume that, for all realizations considered in the theoretical analysis, the following holds:

- $f$  is bounded on  $\mathcal{L}(x_0, \delta_{\max})$ ,
- $f$  and  $\nabla f$  are both  $L$ -Lipschitz continuous on  $\bar{\mathcal{L}}(x_0, \delta_{\max})$ .

**Theorem 3** Let  $\{x_k\}$  be the sequence of iterates generated by the Probabilistic Iterative Hard Thresholding Algorithm (Algorithm 1) under Assumption 1, and moreover assume that the function and iterates are such that Assumption 2 holds. Also assume that the step acceptance parameter  $\eta_2$  satisfies

$$\eta_2 \geq 3\kappa_f \alpha \quad (26)$$

and the function accuracy parameter  $\epsilon_f$  is chosen such that,

$$\epsilon_f \leq \min \{ \kappa_f, \eta_1 \eta_2 \}. \quad (27)$$

Then, if  $\theta$  and  $\beta$  are sufficiently large, it holds that the sequence of trust region bounds  $\{\delta_k\}$  satisfies the summability condition

$$\sum_{k=0}^{\infty} \delta_k^2 < \infty \tag{28}$$

almost surely.

**Proof** We define the constants  $\zeta$  together with  $\nu$  appearing in (25) as satisfying

$$\zeta \geq \max \left\{ a^{-1}, \kappa_g + \max \left\{ \eta_2, \frac{2\epsilon_f + \kappa_f \delta_{max}}{1 - \eta_1} \right\} \right\}, \tag{29}$$

where we recall that

$$a = \frac{1}{2\alpha\kappa_f\delta_{max} + 2\sqrt{K}}$$

and

$$\frac{\nu}{1 - \nu} > \max \left\{ \frac{4\gamma^2}{\zeta c}, \frac{4\gamma^2}{\eta_1\eta_2}, \frac{\gamma^2}{\kappa_f} \right\}, \tag{30}$$

with  $c$  defined by Lemma 3.

We observe that on successful, or accepted, iterations,

$$\Phi_{k+1} - \Phi_k \leq \nu(f(x_{k+1}) - f(x_k)) + (1 - \nu)(\gamma^2 - 1)\delta_k^2 \tag{31}$$

and on unsuccessful iterations,

$$\Phi_{k+1} - \Phi_k \leq (1 - \nu) \left( \frac{1}{\gamma^2} - 1 \right) \delta_k^2 < 0. \tag{32}$$

Let us define the event sequence  $A_k$  as the satisfaction of model accuracy according to Definition 4, that is for all  $y \in B(x_k, \delta_k)$ , the random event occurs in  $\mathcal{F}_k$  such that the realization of  $G_k$  satisfies

$$\|\nabla f(y) - g_k\| \leq \kappa_g \delta_k \quad \text{and} \quad |f(y) - f(x_k) - g_k^T(y - x_k)| \leq \kappa_f \|y - x_k\| \delta_k^2.$$

Furthermore, the sequence of events  $B_k$  is defined as the random event within  $\mathcal{F}_{k+\frac{1}{2}}$  indicating that the function evaluation samples satisfy  $\epsilon_f$ -accuracy according to Definition 3, that is,

$$|f_k^0 - f(x_k)| \leq \epsilon_f \delta_k^2 \quad \text{and} \quad |f_k^s - f(x_k + s_k)| \leq \epsilon_f \delta_k^2.$$

Now fix a realization  $\{\omega_k\}$  for the sequence  $\{X_k, G_k, F_0^k, F_s^k\}$  in  $\mathcal{F}_\infty$  and consider an iterate  $x_k$  in this sequence. We consider the different cases of an approximate stationarity condition denoted as:

$$\|[\nabla f(x_k)]_{I_k}\| \leq \epsilon,$$

**Case 1**  $\|[\nabla f(x_k)]_{I_k}\| \geq \zeta \delta_k$ .

We examine the following subcases based on different events:

- (a)  $A_k \cap B_k$ : The model  $g_k$  satisfies the  $\kappa\text{-}\delta_k$  accuracy condition as well as having  $\varepsilon_f$  accurate function evaluations. Applying (29),

$$\|[\nabla f(x_k)]_{I_k}\| \geq \delta_k/a.$$

Rearranging, we obtain

$$\delta_k \leq a\|[\nabla f(x_k)]_{I_k}\| \leq \frac{a \max\{\delta_k, \alpha\|[\nabla f(x_k)]_{I_k}\|\}}{\alpha}.$$

Notice that this implies (18), that is,

$$\delta_k \leq a \left\| x_k - P_{I_k} \left( x_k - \alpha \min \left\{ 1, \frac{\delta_k}{\alpha\|\nabla f(x_k)\|} \right\} \nabla f(x_k) \right) \right\|,$$

and so we can apply Lemma 3 to conclude that

$$f(x_k) - f(\hat{x}_k) \geq \frac{1}{2\alpha} \|\hat{x}_k - x_k\|^2.$$

Moreover, due to model accuracy it holds that

$$\|g_k\| \geq \|\nabla f(x_k)\| - \kappa_g \delta_k \geq (\zeta - \kappa_g) \delta_k \geq \min \left\{ \frac{1}{\eta_2}, \frac{1 - \eta_1}{2\varepsilon_f + \kappa_f \delta_{max}} \right\} \delta_k.$$

As such, we can apply Lemma 4 to conclude that the step is accepted and Lemma 3 to conclude that the stochastic process proceeds as

$$\begin{aligned} \Phi_{k+1} - \Phi_k &\leq -\nu c \delta_k \left\| x_k - P_{I_k} \left( x_k - \alpha \min \left\{ 1, \frac{\delta_k}{\alpha\|\nabla f(x_k)\|} \right\} \nabla f(x_k) \right) \right\| \\ &\quad + (1 - \nu)(\gamma^2 - 1) \delta_k^2 \\ &\leq [-\nu c \zeta + (1 - \nu)(\gamma^2 - 1)] \delta_k^2 < 0, \end{aligned} \quad (33)$$

where the second inequality uses the case assumption.

- (b)  $A_k \cap B_k^c$ : The function values  $f_k^0, f_k^s$  do not satisfy the  $\varepsilon_f$ -accuracy condition, while model accuracy still holds. In this case the same argument as part (a) holds, with the caveat that erroneous function estimates could lead to a step rejection. In that case, the change in the stochastic process is bounded by (32), that is,

$$\Phi_{k+1} - \Phi_k = (1 - \nu) \left( \frac{1}{\gamma^2} - 1 \right) \delta_k^2 < 0.$$

(c)  $A_k^c \cap B_k$ : If the step is unsuccessful then again we can apply (32). Otherwise, with accurate function estimates, we know from Lemma 4 together with (27) that in this case

$$\Phi_{k+1} - \Phi_k \leq [-\nu\eta_1\eta_2 + (1 - \nu)(\gamma^2 - 1)] \delta_k^2,$$

which is still bounded by (32) on account of (30).

(d)  $A_k^c \cap B_k^c$  In this case, standard Lipschitz arguments give the following bound on the increase in the value of  $\Phi$ :

$$\Phi_{k+1} - \Phi_k \leq \nu C_L \|[\nabla f(x_k)]_{I_k}\| \delta_k + (1 - \nu)(\gamma^2 - 1) \delta_k^2, \quad C_L := \left( 1 + \frac{3L}{2\zeta} \right).$$

We can finally combine these results to obtain, using the definitions of the probabilities  $\theta$  and  $\beta$ ,

$$\begin{aligned} \mathbb{E} [\Phi_{k+1} - \Phi_k | \mathcal{F}_k] &\leq \theta\beta[-\nu c \|[\nabla f(x_k)]_{I_k}\| \delta_k + (1 - \nu)(\gamma^2 - 1) \delta_k] \\ &\quad + [\theta(1 - \beta) + (1 - \theta)\beta](1 - \nu) \left( \frac{1}{\gamma^2} - 1 \right) \delta_k^2 \\ &\quad + (1 - \theta)(1 - \beta) [C_L \|[\nabla f(x_k)]_{I_k}\| \delta_k + (1 - \nu)(\gamma^2 - 1) \delta_k^2]. \end{aligned}$$

We can observe that we can proceed along the same lines as the proof of Case 1 in [19, Theorem 4.11] to conclude that with  $\theta, \beta$  chosen to satisfy

$$\frac{(\theta\beta - 1/2)}{(1 - \theta)(1 - \beta)} \geq \frac{C_L}{c}, \tag{34}$$

we can apply (30) to obtain that both of the following two conditions hold:

$$\mathbb{E} [\Phi_{k+1} - \Phi_k | \mathcal{F}_k, \{ \|[\nabla f(x_k)]_{I_k}\| \geq \zeta\delta_k \}] \leq -\frac{1}{4} c\nu \|\nabla f(x_k)\| \delta_k \tag{35}$$

and

$$\mathbb{E} [\Phi_{k+1} - \Phi_k | \mathcal{F}_k, \{ \|[\nabla f(x_k)]_{I_k}\| \geq \zeta\delta_k \}] \leq -\frac{1}{2}(1 - \nu)(\gamma^2 - 1) \delta_k^2. \tag{36}$$

**Case 2:**  $\|[\nabla f(x_k)]_{I_k}\| < \zeta\delta_k$ .

If  $\|g_k\| < \eta\delta_k$  then (32) holds. Now assume that  $\|g_k\| \geq \eta_2\delta_k$ . We again examine the following subcases based on different events:

- (a)  $A_k \cap B_k$ : The model  $g_k$  satisfies the  $\kappa\delta_k$  accuracy condition as well as having  $\varepsilon_f$  accurate function evaluations. In this case, since it cannot be ensured that the step is accepted, we can apply the argument of Case 1c to conclude that again (32) holds.
- (b)  $A_k \cap B_k^c$ : The function values  $f_k^0, f_k^s$  do not satisfy the  $\varepsilon_f$ -accuracy condition, while model accuracy still holds. An unsuccessful iteration yields (32) a successful iteration satisfies

$$f(x_k) - f(x_{k+1}) = f(x_k) - h_k(x_k) + h_k(x_k) - h_k(\hat{x}_k) + h_k(\hat{x}_k) - f(\hat{x}_k) \geq (\eta_2/\alpha - 2\kappa_f)\delta_k^2 \geq \kappa_f\delta_k^2$$

with (26) responsible for the last inequality. Finally (30) implies (32) holds again.

- (c)  $A_k^c \cap B_k$ : It is the same as Case 1c.
- (d)  $A_k^c \cap B_k^c$ : It is the same as Case 1d.

Now, with  $\theta, \beta$  chosen such that

$$(1 - \theta)(1 - \beta) \leq \frac{(\gamma^2 - 1)(1 - \nu)}{(1 - \nu)(\gamma^4 - 1) + 2\gamma^2 C_L \zeta \nu}, \quad (37)$$

we follow similar arguments to obtain

$$\mathbb{E} [\Phi_{k+1} - \Phi_k | \mathcal{F}_k, \{\|\nabla f(x_k)\|_{I_k} < \zeta\delta_k\}] \leq -\frac{1}{2}(1 - \nu) \left(1 - \frac{1}{\gamma^2}\right) \delta_k^2. \quad (38)$$

Finally, combining the two cases yields that

$$\mathbb{E} [\Phi_{k+1} - \Phi_k | \mathcal{F}_k] \leq -\sigma\delta_k^2$$

with  $\sigma > 0$ , and with the application of standard Martingale Convergence Theory the theorem has been proven.  $\square$

By taking a look at the proof of the previous theorem, it is easy to see that the probabilities involved in the assumptions, i.e.,  $\theta$  and  $\beta$ , should be suitably chosen. This fact is reported in the following remark.

**Remark 1** Following the same reasoning as in [19] Theorem 4.11 and Corollary 4.12, we need to suitably choose  $\theta$  and  $\beta$  so that both equations (34) and (37) are satisfied. By setting those parameters sufficiently large, as suggested in [19], we have that those conditions are satisfied.

We may proceed now to the main and final result. The rest of the original convergence argument can be applied directly to  $\|\nabla f(x_k)\|_{I_k}$ . However, recall that this is not the object that is of primary interest. We are indeed interested in proving that the proposed algorithm gives us a point satisfying some suitable optimality condition with high probability, specifically  $L$ -stationarity expressed in Definition 2.

**Theorem 4** *Let all the assumptions of Theorem 3 hold, with  $\theta$  and  $\beta$  sufficiently large. Then*

$$\lim_{k \rightarrow \infty} \|\nabla f(x_k)_{I_k}\| = 0 \tag{39}$$

almost surely. Moreover, for any limit point  $x^*$  of a realization of iterates  $\{x_k\}$ , we have that there exists some  $\bar{L} > 0$  such that  $x^*$  satisfies  $\bar{L}$ -stationarity, as given by Definition 2.

**Proof** The first part of the statement follows directly from the identical arguments as in [19]. Specifically:

1. From Theorem 3, we have that  $\sum \delta_k^2 < \infty$  holds almost surely, and thus  $\delta_k \rightarrow 0$  almost surely. Then, one can obtain a contradiction to there being some  $\epsilon' > 0$  for which  $\|\nabla f(x_k)_{I_k}\| \geq \epsilon'$ , implying  $\liminf_{k \rightarrow \infty} \|\nabla f(x_k)_{I_k}\| = 0$  as in [19, Theorem 4.16].
2. As in [19, Lemma 4.17], if  $K_\epsilon$  is a subsequence of iterations such that  $\|\nabla f(x_k)_{I_k}\| > \epsilon$  then  $\sum_{k \in K_\epsilon} \delta_k < \infty$  by similarly invoking the condition that  $\|\nabla f(x_k)_{I_k}\| \geq \zeta \delta_k$  and proving that, almost surely,

$$\sum_{k \in K_\epsilon} \Delta_k < \infty. \tag{40}$$

3. Finally, using the previous results, applying the arguments of [19, Theorem 4.18] one can show that, if  $\limsup_{K_\epsilon} \|\nabla f(x_k)_{I_k}\| > \epsilon$ , then  $\sum_{K_\epsilon} \delta_k = \infty$ , obtaining a contradiction with the previous result. This establishes (39).

Consider now an almost sure realization and let  $x^*$  be a limit point of  $\{x_k\}$ , that is, there exists a subsequence  $\{x_k\}_S \rightarrow x^*$ , with  $S \subseteq \{0, 1, \dots\}$ . Now fix the realization and corresponding subsequence for the remainder of the proof. Since  $\sigma_k$  and  $I_k$  are subsets of the finite set  $\{1, \dots, n\}$  for any  $k$ , without loss of generality we can assume that they are constant over the considered subsequence  $S$  (passing into a further subsequence if needed). Namely, after discarding an appropriate finite sequence from the beginning of  $S$ ,

$$I_k = I^* \quad \forall k \in S, \tag{41}$$

$$\sigma_k = \sigma^* \quad \forall k \in S. \tag{42}$$

Observe that, for any  $k$ , we can write

$$\alpha \min \left\{ 1, \frac{\delta_k}{\alpha \|g_k\|} \right\} \|g_k\| \begin{cases} = 0 & \text{if } \|g_k\| = 0, \\ \leq \frac{\alpha \delta_k \|g_k\|}{\alpha \|g_k\|} = \delta_k & \text{if } \|g_k\| > 0. \end{cases}$$

Taking into account that  $\{\delta_k\} \rightarrow 0$  from Theorem 3, we get

$$\lim_{k \rightarrow \infty} \alpha \min \left\{ 1, \frac{\delta_k}{\alpha \|g_k\|} \right\} \|g_k\| = 0.$$

Thus,

$$\lim_{k \rightarrow \infty, k \in S} x_k - \alpha \min \left\{ 1, \frac{\delta_k}{\alpha \|g_k\|} \right\} \|g_k\| = x^*. \quad (43)$$

Since, according to (42) and the definition of  $\sigma_k$  given in (5), we have

$$\sigma^* \in \tilde{\Sigma} \left( x_k - \alpha \min \left\{ 1, \frac{\delta_k}{\alpha \|g_k\|} \right\} g_k \right) \quad \forall k \in S,$$

it follows from (43) that

$$\sigma^* \in \tilde{\Sigma}(x^*).$$

Hence, recalling (41) and the definition of  $I_k$  given in (6), it holds that

$$i \in I_{\mathcal{I}}(x^*) \quad \Rightarrow \quad i \in \mathcal{I}^*, \quad (44)$$

where we have used the fact that  $x^*$  is feasible (and then,  $x_i^* \neq 0$  implies that the  $i$ th component is one of the  $K$  largest ones in  $x^*$ ). Using (44) and (39), it follows that

$$[\nabla f(x^*)]_i = 0 \quad \forall i \in I_{\mathcal{I}}(x^*).$$

We have thus proven that  $x^*$  satisfies Basic Feasibility, according to Definition 1.

We now consider two different cases:

- $\nabla f(x^*) = 0$ . In this case it is easy to see that the point is  $\bar{L}$ -stationary for any choice of  $\bar{L} > 0$ .
- $\nabla f(x^*) \neq 0$ . Recalling Lemma 1 and reasoning as in [3, Remark 2.3], we have that  $\bar{L}$ -stationarity holds with

$$\bar{L} = \max_{i \in I_{\mathcal{A}}(x^*)} \frac{|[\nabla f(x^*)]_i|}{M_K(x^*)}. \quad \square$$

## 5 Numerical results

In this section, we present two machine learning applications of Algorithm 1: adversarial attacks on neural networks and the reconstruction of sparse Gaussian graphical models. The implementation was carried out using the Python program-

ming language, using the NumPy, Keras, Tensorflow, scikit-learn, and Pandas libraries. The hyperparameters were selected as follows:  $\eta_1 = 10^{-4}$ ,  $\eta_2 = 10^{-4}$ ,  $\delta_0 = 1$ ,  $\delta_{\max} = 10$ , and  $\gamma = 2$ . All the experiments were conducted on a machine equipped with an 11th Gen Intel(R) Core(TM) i7-1165G7 CPU @ 2.80GHz (1.69 GHz). The code is available at [https://github.com/Berga53/Probabilistic\\_iterative\\_hard\\_thresholding](https://github.com/Berga53/Probabilistic_iterative_hard_thresholding).

### 5.1 Adversarial attacks on neural networks

Adversarial attacks are techniques used to craft imperceptible perturbations that, when added to regular data inputs, induce misclassifications in neural network models. These perturbations are typically designed to evade human detection while successfully fooling the model’s classification process. One of the most powerful type of adversarial attack is the Carlini and Wagner [29], characterized by the following formulation:

$$\begin{aligned} \min_{\delta} D(x, x + \Delta) + c \cdot f(x + \Delta) \\ \text{such that } x + \Delta \in [0, 1]^n, \end{aligned} \tag{45}$$

with  $\Delta$  being the perturbation,  $D$  being usually the  $\ell_2$  or  $\ell_0$  distance, and

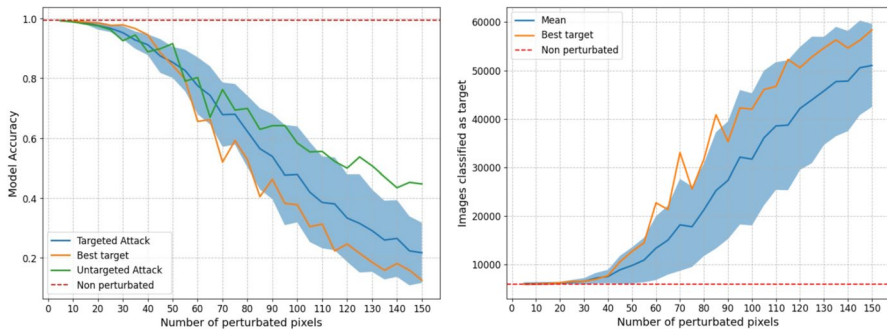
$$f(x) = \left( \max_{i \neq t} ([F(x)]_i) - [F(x)]_t \right)^+, \tag{46}$$

where  $[F(x)]_i$  is the probability output for the class  $i$ , and  $t$  is the targeted class.

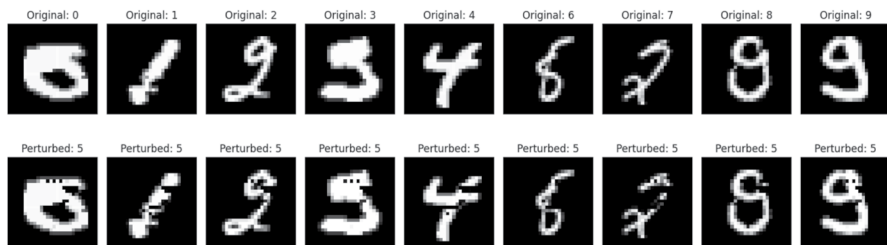
Using our algorithm, we can incorporate the  $\ell_0$  penalty directly in the constraint, so our final formulation of the problem is

$$\begin{aligned} \min_{\|\delta\|_0 \leq K} \|\Delta\|_2 + c \cdot f(x + \Delta) \\ \text{such that } x + \Delta \in [0, 1]^n. \end{aligned} \tag{47}$$

In practice, this allows us to decide how many pixels to perturb during the attack. While usual attacks are trained against selected samples of the dataset, in this paper, we will demonstrate a universal adversarial attack: the attack is performed against the entirety of the dataset, producing only one global perturbation. We will show that, in both targeted and untargeted attacks, we can significantly lower a model’s accuracy using very few pixels. We tested the attack on the MNIST dataset, which consists of 60,000 images of handwritten digits (0-9) that are  $28 \times 28$  pixels in size. We performed both targeted and untargeted attacks. In the targeted attack, we aimed to misclassify the images into a specific class, using a different digit as target class in different experiments. This approach allows us to manipulate the model to misclassify any digit as any chosen target digit. In the untargeted attack, we simply aimed to cause any misclassification, choosing, for every sample, the easiest class to target. However, the untargeted attack is generally a bit weaker in the context of the Car-



**Fig. 1** Effect of increasing the sparsity constraint on accuracy and targeted attack predictions



**Fig. 2** Example of perturbed images with  $\|\delta\|_0 = 25$  and target 5

lini and Wagner Attack. We will show that, in both targeted and untargeted attacks, we can significantly lower a model's accuracy using very few pixels. We gradually increase the sparsity constraint and observe that this gradually increases the errors made by the model. In particular, in Figure 1, we can see both the accuracy decreasing and the number of samples predicted as the attack target increasing, indicating that the attack is performed as desired. We present the mean and variance of the different experiments in blue, the best-targeted attack in orange, and the untargeted attack in green. As expected, the untargeted attack is weaker than the targeted one. In Figure 2, we observe examples of both the original and perturbed images where the attacks were successful, specifically targeting the digit 5.

## 5.2 Sparse Gaussian graphical models

Probabilistic Graphical Models are a popular tool in machine learning to model the relationships between random variables. The Gaussian Graphical Model is an undirected graph with each edge corresponding to a Gaussian conditional probability of one variable at the end of the edge to another. By learning the adjacency matrix together with the model weights, we can infer the proximal physical and possibly causal relationships between quantities.

This is of special importance in high dimensional settings (see, e.g., [30]). Whereas in many contemporary “big data” approaches the sample size is many orders of magnitudes larger than the dimensionality of feature space, there are a number of settings

wherein obtaining data samples is costly, and such a regime cannot be expected to hold. Indeed this is often the case in medical applications, wherein recruiting volunteers for a clinical trial, or even obtaining health records, presents formidable costs to significant scaling in sample size. On the other hand, the precision of instrumentation has led to detailed personal physiological and biomarker data, yielding a very high dimensional feature space. One associated observation is that in the underdetermined case, when the dimensionality of the features exceeds the number of samples, some of the guarantees associated with the  $\ell_1$  proxy for sparsity are no longer applicable, bringing greater practical salience to having a reliable algorithm enforcing sparsity explicitly.

The recent work [23] presented an integer programming formulation for training sparse Gaussian graphical models. Prior to redefining the sparsity regularization using binary variables, their  $\ell_0$  optimization problem is given as

$$\min_{W \in \mathbb{S}^p} F_0(W) := \sum_{i=1}^p \left( -\log(w_{ii}) + \frac{1}{w_{ii}} \|\tilde{X} w_i\|^2 \right) + \lambda_0 \|W\|_0 + \lambda_2 \|W\|_2^2, \tag{48}$$

with  $\tilde{X} = \frac{1}{\sqrt{n}} X$  the scaled feature matrix and  $X \in \mathbb{R}^{p \times n}$  a matrix consisting of  $p$  measures and  $n$  samples,  $W \in \mathbb{S}^p$  the weight matrix related to the graph,  $\mathbb{S}^p$  being the set of symmetric matrices in  $\mathbb{R}^{p \times p}$ ,  $w_{ii}$  indicating the  $i$ -th component in the diagonal of  $W$  and  $w_i$  indicating the  $i$ -th row of  $W$ . Functionally,  $w_{ij}$  defines an edge between node  $i$  and  $j$  in the graph, with a nonzero indicating the presence of an active edge, which corresponds to a direct link in the perspective of DAG structure of the group. The value associated with the edge corresponds to the weight defining the strength of the interaction between the features  $i$  and  $j$ . We seek to regularize cardinality for the sake of encouraging parsimonious models, as well as minimizing the total norm of the weights for general regularization.

Due to the structure of our algorithm, we can modify the formulation of the problem by incorporating the  $\ell_0$  constraint. The final formulation of the problem is then expressed as follows:

$$\min_{W \in \mathbb{S}^p, \|W\|_0 \leq K} F_0(W) := \sum_{i=1}^p \left( -\log(w_{ii}) + \frac{1}{w_{ii}} \|\tilde{X} w_i\|^2 \right) + \lambda_2 \|W\|_2^2. \tag{49}$$

We also observed that the  $\ell_0$  constraint in our formulation is very strong. In practical applications, we eliminate  $\lambda_2$  penalty term, as the  $\ell_0$  constraint was the dominant factor in the model.

We applied the model to the GDS2910 dataset from the Gene Expression Omnibus (GEO). This dataset consists of gene expression profiles, which naturally yield a high-dimensional feature space, with 1900 features and 191 samples. Given this feature-to-sample ratio, we can assume some level of sparsity in the final adjacency matrix. Since there is no ground truth for the underlying structure, our goal is to investigate how changing the  $\ell_0$  constraint affects the results of our method, while also gathering information on the true sparsity nature of the data. We performed the

test by gradually increasing  $K$ , the  $\ell_0$  constraint, from 5000 to 15,000. This range was previously determined to be optimal based on preliminary tests. Note that the adjacency matrix we are searching for is of size  $1900 \times 1900$ , resulting in a total of  $3.61 \cdot 10^6$  entries. To ensure the robustness of the results, for each value of  $K$ , we performed ten runs starting from different randomly chosen feasible points, and the algorithm was given a total of 1000 iteration for every run. We also decided to set the  $\lambda_2$  parameter to zero, as we observed that the strong  $\ell_0$  constraint was dominant over the  $\ell_2$  penalty.

We also divided the dataset into training and validation sets to determine whether the reconstructed matrix is a result of overfitting. In Fig. 3, we show the effect of varying  $K$ , which represents the number of nonzero entries that the matrix is allowed to have. The figure on the left, which shows the average objective value found over the ten runs, demonstrates that increasing  $K$  eventually stops being beneficial to the model's performance. Additionally, we observe that the number of mean accepted iterations also stops increasing, indicating that the model cannot extract more information from the data. This suggests that the true sparsity of the data can be estimated by identifying the point at which further increasing  $K$  no longer improves the model's results. In Fig. 4, we present an example from our tests where the objective function decreases over the successful iterations.

## 6 Conclusions

In this paper, we addressed the stochastic cardinality-constrained optimization problem, providing a well defined algorithm, convergence theory and illustrative experiments. Many contemporary machine learning applications involve scenarios where sparsity is crucial for high-dimensional model fitting. We proposed an iterative hard-thresholding like algorithm based on probabilistic models that nicely balances computational efficiency and solution precision by allowing flexible gradient estimates while incorporating hard sparsity constraints.

We analyzed the theoretical properties of the method and proved almost sure convergence to L-stationary points under mild assumptions. This extends previous work in the optimization literature on finding solutions with strong stationarity guarantees

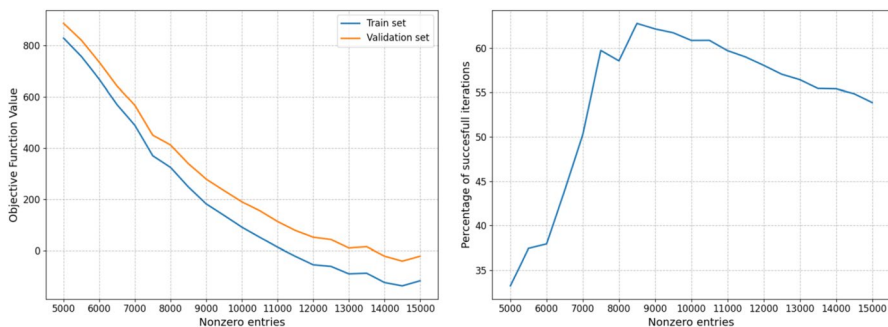
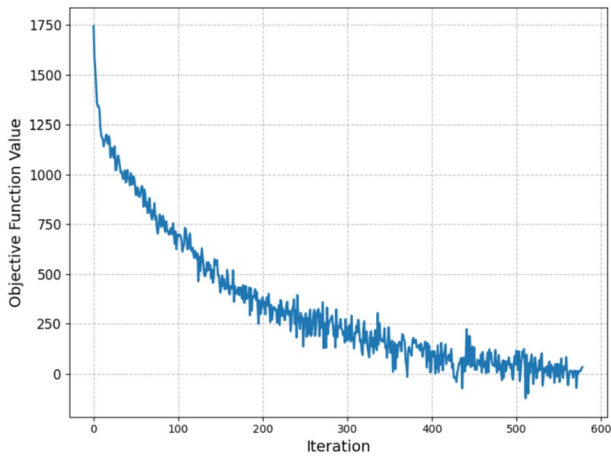


Fig. 3 Effect of increasing the sparsity constraint  $K$



**Fig. 4** Objective function over the iterations

together with machine learning articles that perform iterative hard thresholding with stochastic gradients to achieve a novel balance between ease of a fast implementation and formal guarantees of performance. The numerical experiments confirmed the practical effectiveness of our method, showcasing its potential in machine learning tasks such as adversarial attacks and probabilistic graphical model training. By enforcing explicit cardinality constraints, our approach was able to produce models with enhanced sparsity and interpretability in the end.

Future work may involve extending the algorithm to accommodate additional nonlinear constraints, exploring techniques to further improve scalability and performance, as well as testing the algorithm on some other relevant Machine Learning applications, like, e.g., sparse Dynamic Bayesian Network training.

**Funding** The work of Vyacheslav Kungurtsev was funded by the European Union’s Horizon Europe research and innovation programme under grant agreement No. 101084642.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Lämmel, S., Shikhman, V.: On nondegenerate  $m$ -stationary points for sparsity constrained nonlinear optimization. *J. Global Optim.* **82**(2), 219–242 (2022)

2. Lämmel, S., Shikhman, V.: Critical point theory for sparse recovery. *Optimization* **72**(2), 521–549 (2023)
3. Beck, A., Eldar, Y.C.: Sparsity constrained nonlinear optimization: optimality conditions and algorithms. *SIAM J. Optim.* **23**(3), 1480–1509 (2013)
4. Beck, A., Hallak, N.: On the minimization over sparse symmetric sets: projections, optimality conditions, and algorithms. *Math. Oper. Res.* **41**(1), 196–223 (2016)
5. Hallak, N.: A path-based approach to constrained sparse optimization. *SIAM J. Optim.* **34**(1), 790–816 (2024)
6. Branda, M., Bucher, M., Červinka, M., Schwartz, A.: Convergence of a Scholtes-type regularization method for cardinality-constrained optimization problems with an application in sparse robust portfolio optimization. *Comput. Optim. Appl.* **70**(2), 503–530 (2018)
7. Bucher, M., Schwartz, A.: Second-order optimality conditions and improved convergence results for regularization methods for cardinality-constrained optimization problems. *J. Optim. Theory Appl.* **178**(2), 383–410 (2018)
8. Burdakov, O., Kanzow, C., Schwartz, A.: Mathematical programs with cardinality constraints: reformulation by complementarity-type conditions and a regularization method. *SIAM J. Optim.* **26**(1), 397–425 (2016)
9. Červinka, M., Kanzow, C., Schwartz, A.: Constraint qualifications and optimality conditions for optimization problems with cardinality constraints. *Math. Program.* **160**(1), 353–377 (2016)
10. Lapucci, M., Levato, T., Sciacrone, M.: Convergent inexact penalty decomposition methods for cardinality-constrained problems. *J. Optim. Theory Appl.* **188**(2), 473–496 (2021)
11. Lapucci, M., Levato, T., Rinaldi, F., Sciacrone, M.: A unifying framework for sparsity-constrained optimization. *J. Optim. Theory Appl.* **199**(2), 663–692 (2023)
12. Lu, Z., Zhang, Y.: Sparse approximation via penalty decomposition methods. *SIAM J. Optim.* **23**(4), 2448–2478 (2013)
13. Zhou, P., Yuan, X., Feng, J.: Efficient stochastic gradient hard thresholding. *Adv. Neural Inform. Process. Syst.* **31** (2018)
14. Zhou, B., Chen, F., Ying, Y.: Stochastic iterative hard thresholding for graph-structured sparsity optimization. In: *International Conference on Machine Learning*, pp. 7563–7573 (2019). PMLR
15. Murata, T., Suzuki, T.: Sample efficient stochastic gradient iterative hard thresholding method for stochastic sparse linear regression with limited attribute observation. *Adv. Neural Inform. Process. Syst.* **31** (2018)
16. Jain, P., Tewari, A., Kar, P.: On iterative hard thresholding methods for high-dimensional m-estimation. *Adv. Neural Inform. Process. Syst.* **27** (2014)
17. Bastin, F., Cirillo, C., Toint, P.L.: An adaptive monte Carlo algorithm for computing mixed logit estimators. *CMS* **3**, 55–79 (2006)
18. Bandeira, A.S., Scheinberg, K., Vicente, L.N.: Convergence of trust-region methods based on probabilistic models. *SIAM J. Optim.* **24**(3), 1238–1264 (2014)
19. Chen, R., Menickelly, M., Scheinberg, K.: Stochastic optimization using a trust-region method and random models. *Math. Program.* **169**, 447–487 (2018)
20. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57 (2017). IEEE
21. Croce, F., Hein, M.: Sparse and imperceivable adversarial attacks. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4724–4732 (2019)
22. Modas, A., Moosavi-Dezfooli, S.-M., Frossard, P.: Sparsefool: a few pixels make a big difference. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9087–9096 (2019)
23. Behdin, K., Chen, W., Mazumder, R.: Sparse gaussian graphical models with discrete optimization: computational and statistical perspectives. Preprint at [arXiv:2307.09366](https://arxiv.org/abs/2307.09366) (2023)
24. Negri, M.M., Arend Torres, F., Roth, V.: Conditional matrix flows for gaussian graphical models. *Adv. Neural. Inf. Process. Syst.* **36**, 25095–25111 (2023)
25. Kanzow, C., Raharja, A.B., Schwartz, A.: Sequential optimality conditions for cardinality-constrained optimization problems with applications. *Comput. Optim. Appl.* **80**, 185–211 (2021)
26. Blumensath, T., Davies, M.E.: Iterative thresholding for sparse approximations. *J. Fourier Anal. Appl.* **14**, 629–654 (2008)
27. Bertsekas, D.P.: *Nonlinear Programming*. Athena Scientific, Belmont (1999)
28. Cartis, C., Scheinberg, K.: Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. *Math. Program.* **169**, 337–375 (2018)

29. Carlini, N., Wagner, D.A.: Towards evaluating the robustness of neural networks. Preprint at [arXiv:1608.04644](https://arxiv.org/abs/1608.04644) (2016)
30. Wainwright, M.J.: High-Dimensional Statistics: A Non-Asymptotic Viewpoint. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge (2019)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.