Greedy and IHT Algorithms for Non-convex Optimization with Monotone Costs of Non-zeros

Shinsaku Sakaue
NTT Communication Science Laboratories

Abstract

Non-convex optimization methods, such as greedy-style algorithms and iterative hard thresholding (IHT), for ℓ_0-constrained minimization have been extensively studied thanks to their high empirical performances and strong guarantees. However, few works have considered non-convex optimization with general non-zero patterns; this is unfortunate since various non-zero patterns are quite common in practice. In this paper, we consider the case where non-zero patterns are specified by monotone set functions. We first prove an approximation guarantee of a cost-benefit greedy (CBG) algorithm by using the *weak submodularity* of the problem. We then consider an IHT-style algorithm, whose projection step uses CBG, and prove its convergence guarantee. We also provide many applications and experimental results that confirm the advantages of the algorithms introduced.

1 INTRODUCTION

In many practical optimization problems, target solutions are expected to have certain non-zero patterns such as contiguous sparsity and group sparsity to name a few. In the context of convex optimization approaches (Tibshirani et al., 2005; Bach, 2010; Tewari et al., 2011; Chen et al., 2012), such problems are often formulated with regularization terms that represent certain structures of non-zeros; e.g., fused lasso, group lasso, and submodularity-based regularization.

For the case where only the number of non-zeros is constrained, i.e., ℓ_0-constrained minimization, non-convex optimization algorithms such as greedy-style algorithms, iterative hard thresholding (IHT), and hard thresholding pursuit (HTP) have recently been attracting much attention thanks to their strong theoretical guarantees and high empirical performances (Jain et al., 2014; Yuan et al., 2016; Elenberg et al., 2018). Most of the existing works are built on desirable properties of the ℓ_0-constraint (e.g., exact projection is possible), and thus to address non-convex optimization with more general constraints on non-zeros is very challenging. Therefore, only a few works have studied non-convex optimization methods that can deal with more general constraints than the ℓ_0-constraint, even though various non-zero patterns can arise in practice as considered in the context of convex optimization. Namely, the field still awaits a theoretical analysis of non-convex optimization methods that can handle problems with a wide variety of constraints on non-zeros.

In this paper, we consider non-convex constrained optimization problems formulated as follows:

$$\underset{\mathbf{x}\in\mathbb{R}^{[d]}}{\text{minimize}}\ \ l(\mathbf{x}) \quad \text{subject to } G(\text{supp}(\mathbf{x})) \le c, \quad (1)$$

where $[d] := \{1, \dots, d\}$ and $\text{supp}(\mathbf{x}) \subseteq [d]$ is the *support* of $\mathbf{x}$, which is the set of indices corresponding to the non-zeros of $\mathbf{x}$. We assume that the objective function, $l : \mathbb{R}^{[d]} \to \mathbb{R}$, is continuously differentiable and that it has *restricted strong convexity* (RSC) and *restricted smoothness* (RSM) as detailed later; $l(\cdot)$ may be non-convex on $\mathbb{R}^{[d]}$ in general. Set function $G : 2^{[d]} \to \mathbb{R}$, which we call the *cost function*, is monotone and normalized (i.e., $G(\emptyset) = 0$), and $c > 0$ represents a budget value; the resulting feasible region is generally non-convex. We assume that $G(\cdot)$ has a positive *restricted inverse curvature* as detailed later. We define $\mathcal{F} := \{\mathsf{S} \subseteq [d] \mid G(\mathsf{S}) \le c\} \subseteq 2^{[d]}$, which is the collection of all feasible supports. Note that the ℓ_0-constrained minimization, whose constraint is given by $G(\text{supp}(\mathbf{x})) = |\,\text{supp}(\mathbf{x})| =: \|\mathbf{x}\|_0$, is a special case of problem (1). As elucidated in Section 5, optimization problems that can be formulated as (1) arise in many applications. RSC and RSM, imposed on the objective function, have been used in many recent studies since they are useful for deriving theoretical guarantees of

Proceedings of the 22$^{\text{nd}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89. Copyright 2019 by the author(s).

non-convex optimization methods. As in (Jain et al., 2014), RSC/RSM-based analysis is advantageous in that it requires neither the demanding *restricted isometric property* (RIP) condition (Candès et al., 2006) nor strong convexity (SC) and smoothness (SM) over the whole domain, $\mathbb{R}^{[d]}$.

1.1 Our Contributions

We make the following contributions:

- We consider a cost-benefit greedy (CBG) for problem (1) and prove its approximation guarantee based on (Elenberg et al., 2018), which connects RSC/RSM to *weak submodularity*.
- We consider an IHT-style method for problem (1), whose projection step is performed with CBG, and prove its convergence guarantee by using the above approximation guarantee of CBG.
- An advantage of the above CBG and IHT is that they can be applied to a wide variety of non-convex constrained optimization problems of form (1). To demonstrate this, we provide many examples of cost functions, $G(\cdot)$.
- We experimentally evaluate CBG and IHT. Specifically, we show that IHT is advantageous when problems are *well-conditioned*, while CBG is effective for *ill-conditioned* problems. We also confirm that both CBG and IHT are robust against the non-convexity of objective functions compared to an alternative convex optimization method. We finally confirm the practical utility of CBG and IHT via experiments on real-world data.

1.2 Related work

Our work is related to greedy and IHT-style algorithms. Below we show some relevant studies on each algorithm.

Greedy Algorithm In the area of sparse optimization, greedy-style algorithms such as orthogonal matching pursuit (OMP) (Pati et al., 1993), CoSaMP (Needell and Tropp, 2009), and forward greedy selection (Shalev-Shwartz et al., 2010) have been studied extensively. For optimization problems whose constraints are defined with subadditive *coding complexity*, Huang et al. (2011) developed a greedy-style algorithm called *StructOMP*. Unlike these works, our analysis of CBG accepts constraints defined with monotone set functions. Most of the above studies rely on the RIP condition (Candès et al., 2006) or other similar assumptions; the RIP condition is somewhat demanding since it requires the objective function to be quadratic and its Hessian to have a bounded *condition number* on some restricted space. Jain et al. (2014) showed that RSC/RSM provide a more general analysis framework for non-convex sparse optimization; after that, many RSC/RSM-based guarantees have been studied for ℓ_0-constrained minimization (Yuan et al., 2016; Jain and Kar, 2017). Recently, it has been revealed that ℓ_0-constrained minimization with RSC/RSM objective functions can be seen as weakly submodular maximization under a cardinality constraint (Liberty and Sviridenko, 2017; Elenberg et al., 2018), implying that greedy maximization of weakly submodular function is a promising approach to non-convex optimization with RSC/RSM objectives. However, few studies have considered weakly submodular maximization with constraints that are more general than the cardinality constraint. The only exception is (Chen et al., 2018), which studies weakly submodular maximization under a matroid constraint. In contrast, our work considers weakly submodular maximization under a monotone set-function constraint, which cannot be represented as a matroid constraint in general. (Non-)submodular maximization problems with various constraints were studied in (Zhang and Vorobeychik, 2016; Qian et al., 2018), but their objective functions are not proved to have connection to RSC/RSM objectives unlike weakly submodular functions.

IHT-style Algorithm For non-convex sparse optimization, IHT/HTP-style methods have been widely studied (Blumensath and Davies, 2009; Foucart, 2011). Recently, RSC/RSM-based guarantees of IHT/HTP-style methods have been proved for ℓ_0-constrained minimization (Jain et al., 2014; Yuan et al., 2016). IHT-style methods for non-convex optimization problems with other constraints have also been studied, but the problem settings considered in the existing works are different from problem (1). For example, Khanna and Kyrillidis (2018) have developed an accelerated IHT for the case where exact projection onto the feasible region is possible. Barber and Ha (2018) have proved convergence of projected gradient descent for the case where the feasible region is parametrized with *local concavity*. Algorithms based on *head* and *tail approximations* have been studied for non-convex optimization defined on graph structures (Hegde et al., 2015; Zhou and Chen, 2016). Examples of problem (1) include a variant of this setting as in Section 5, but our algorithm is different from the existing methods in that it can be applied to other various problems and that it requires only simple greedy projection and a gradient descent procedure, while the existing methods rely on an elaborated approximation algorithm for an NP-hard problem, called the *prize-collecting Steiner tree*. The closest to our method is an IHT-style method with greedy projection for non-convex optimization with group sparsity (Jain et al., 2016). The difference between the two methods is as follows: In the projection step of (Jain et al., 2016), the greedy algorithm approximately solves a submodular maximization problem under a cardinality constraint; hence the well-known analysis (Nemhauser et al., 1978) is applicable. On the other hand, in our IHT with CBG projection, the CBG algorithm aims to solve modular function maximization under a monotone set-function constraint; to guarantee the performance of this projection step, we need an approximation guarantee of CBG, which is our first contribution and is presented in Section 3.

1.3 Organization

Section 2 provides necessary definitions and background. Our main results on CBG and IHT are presented in Sections 3 and 4, respectively. Section 5 presents examples of monotone cost functions. Experiments are shown in Section 6. Section 7 concludes this paper. All proofs are presented in the appendices.

2 BACKGROUND

Below we introduce the definitions and background that are necessary for the subsequent discussion.

Sets and Set Functions Subsets of $[d]$ are denoted by upper case sans-script fonts: e.g., S and T. Elements in $[d]$ are basically denoted by j; we sometimes abuse the notation and denote $\{j\} \subseteq [d]$ simply by j. Set functions defined on $2^{[d]}$ are denoted by upper case letters: e.g., F and G. Given set function $F : 2^{[d]} \to \mathbb{R}$, we define $F(\mathsf{T} \mid \mathsf{S}) := F(\mathsf{S} \cup \mathsf{T}) - F(\mathsf{S})$ for any $\mathsf{S}, \mathsf{T} \subseteq [d]$. All set functions considered in this paper are monotone: $F(\mathsf{T} \mid \mathsf{S}) \geq 0$ for any $\mathsf{S}, \mathsf{T} \subseteq [d]$. We say $F(\cdot)$ is *submodular* if it satisfies $F(j \mid \mathsf{S}) \geq F(j \mid \mathsf{T})$ for any $\mathsf{S} \subseteq \mathsf{T}$ and $j \notin \mathsf{T}$ and *supermodular* if it satisfies $F(j \mid \mathsf{S}) \leq F(j \mid \mathsf{T})$ for any $\mathsf{S} \subseteq \mathsf{T}$ and $j \notin \mathsf{T}$.

Submodularity Ratio Given monotone $F : 2^{[d]} \to \mathbb{R}$, its weak submodularity is parametrized with the following *submodularity ratio* (Das and Kempe, 2011). Let $\mathsf{U} \subseteq [d]$ and $k > 0$ be a fixed subset and integer, respectively. We define submodularity ratio $\gamma_{\mathsf{U},k}$ of $F(\cdot)$ as the largest scalar that satisfies

$$\gamma_{\mathsf{U},k} F(\mathsf{S} \mid \mathsf{L}) \leq \sum_{j \in \mathsf{S}} F(j \mid \mathsf{L})$$

for any disjoint $\mathsf{L}, \mathsf{S} \subseteq [d]$ such that $\mathsf{L} \subseteq \mathsf{U}$ and $|\mathsf{S}| \leq k$. We have $\gamma_{\mathsf{U},k} \in [0, 1]$ for any U and k. In particular, we have $\gamma_{\mathsf{U},k} = 1$ iff $F(\cdot)$ is submodular. Note that $\gamma_{\mathsf{U},k} \leq \gamma_{\mathsf{U}',k'}$ holds for any $\mathsf{U}' \subseteq \mathsf{U}$ and $k' \leq k$.

Superadditivity Ratio Let $G : 2^{[d]} \to \mathbb{R}$ be any monotone set function and $k > 0$ be a fixed integer. As in (Bogunovic et al., 2018), we define *superadditivity ratio* β_k of $G(\cdot)$ as the largest scalar that satisfies

$$\beta_k \sum_{j \in \mathsf{S}} G(j) \leq G(\mathsf{S})$$

for any $|\mathsf{S}| \leq k$. Note that we have $\beta_k \leq \beta_{k'}$ for any $k' \leq k$ and that $\beta_k \in [1/k, 1]$ holds due to monotonicity. If $G(\cdot)$ is supermodular, we have $\beta_k = 1$.

Curvature and Inverse Curvature Given a monotone set function $G : 2^{[d]} \to \mathbb{R}$, generalized curvature (Bian et al., 2017) and generalized inverse curvature (Bogunovic et al., 2018) of G are defined as the smallest scalars $\alpha, \check{\alpha} \in [0, 1]$, respectively, that satisfy

$$G(j \mid \mathsf{S}\backslash\{j\} \cup \mathsf{M}) \geq (1 - \alpha) G(j \mid \mathsf{S}\backslash\{j\})$$
$$G(j \mid \mathsf{S}\backslash\{j\}) \geq (1 - \check{\alpha}) G(j \mid \mathsf{S}\backslash\{j\} \cup \mathsf{M})$$

for any $\mathsf{S}, \mathsf{M} \subseteq [d]$ and $j \in \mathsf{S}\backslash\mathsf{M}$. Function G is submodular (supermodular) iff $\check{\alpha} = 0$ ($\alpha = 0$). In our analysis, it suffices that a variant of the generalized inverse curvature of cost function G is bounded, which we define as the largest scalar $\theta \in [0, 1]$ that satisfies

$$G(j) \geq \theta G(j \mid \mathsf{M})$$

for any $\mathsf{M} \subseteq [d]$ and $j \notin \mathsf{M}$. We refer to θ as *restricted inverse curvature* since it is defined by restricting the definition of $\check{\alpha}$ to the case where S is a singleton: $\mathsf{S} = \{j\}$. Note that we have $\theta \geq 1 - \check{\alpha}$.

Vectors and Matrices Vectors are denoted by bold lower case letters (e.g., $\mathbf{x}$ and $\mathbf{y}$); zero vectors are denoted simply by 0. Given $\mathsf{S} \subseteq [d]$ and $\mathbf{x} \in \mathbb{R}^{[d]}$, whose j-th entry is associated with $j \in [d]$, $\mathbf{x}_\mathsf{S} \in \mathbb{R}^\mathsf{S}$ denotes the restriction of $\mathbf{x}$ to S. Matrices are denoted by bold upper case letters (e.g., $\mathbf{A}$ and $\mathbf{X}$); $\mathbf{I}$ denotes the identity matrix. Given $\mathbf{A} \in \mathbb{R}^{[n] \times [d]}$, where n is a positive integer, $\mathbf{A}_\mathsf{S} \in \mathbb{R}^{[n] \times \mathsf{S}}$ denotes a submatrix whose columns are restricted to S. Given square matrix $\mathbf{A} \in \mathbb{R}^{[d] \times [d]}$, $\mathbf{A}_{\mathsf{S},\mathsf{S}} \in \mathbb{R}^{\mathsf{S} \times \mathsf{S}}$ denotes a square submatrix whose rows and columns are restricted to S.

Restricted Strong Convexity and Restricted Smoothness Given a continuously differentiable function, $l : \mathbb{R}^{[d]} \to \mathbb{R}$, and $\Omega \subseteq \mathbb{R}^{[d]} \times \mathbb{R}^{[d]}$, we say $l(\cdot)$ is μ_Ω-RSC and ν_Ω-RSM if it satisfies

$$\begin{aligned} l(\mathbf{y}) &\geq l(\mathbf{x}) + \langle \nabla l(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{\mu_\Omega}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 \\ l(\mathbf{y}) &\leq l(\mathbf{x}) + \langle \nabla l(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{\nu_\Omega}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 \end{aligned} \tag{2}$$

for all $(\mathbf{x}, \mathbf{y}) \in \Omega$, where $\|\cdot\|_2$ denotes the ℓ_2-norm. We refer to μ_Ω and ν_Ω as RSC and RSM constants, respectively. Note that, given μ_Ω, ν_Ω, and $\Omega' \subseteq \Omega$, we can set $\mu_{\Omega'}$ and $\nu_{\Omega'}$ so as to satisfy $\mu_{\Omega'} \geq \mu_\Omega$ and $\nu_{\Omega'} \leq$

ν_Ω, respectively. We call $\kappa_\Omega \coloneqq \nu_\Omega/\mu_\Omega \geq 1$ a *restricted condition number* (Jain and Kar, 2017), and objective functions with a smaller restricted condition number are typically easier to deal with. For convenience, if (2) holds with $\Omega = \{(\mathbf{x}, \mathbf{y}) \mid \|\mathbf{x}\|_0 \leq k_1, \|\mathbf{y}\|_0 \leq k_1, \|\mathbf{x}-\mathbf{y}\|_0 \leq k_2\}$, we say $l(\cdot)$ is μ_{k_1,k_2}-RSC and ν_{k_1,k_2}-RSM. Furthermore, we define $\mu_k \coloneqq \mu_{k,k}$ and $\nu_k \coloneqq \nu_{k,k}$.

Other Setups In what follows, we define $\rho \coloneqq \max_{j\in[d],\mathsf{S}\subseteq[d]} G(j \mid \mathsf{S})$; i.e., ρ is the maximum marginal increase of G yielded by adding a single element. As in (Shalev-Shwartz et al., 2010), in the context of ℓ_0-constrained minimization, whose constraint is given by $\|\mathbf{x}\|_0 \leq k$, we typically have a trade-off between sparsity k and objective error $\ell(\mathbf{x}) - \ell(\mathbf{x}^*)$, where $\mathbf{x}^* \in \mathbb{R}^{[d]}$ is a target sparse solution. Considering this background, most existing guarantees are parametrized with k and $k^* = \|\mathbf{x}^*\|_0$. Since problem (1) includes the ℓ_0-constrained minimization, similar parametrization is naturally needed to obtain theoretical guarantees for problem (1). Here, we fix $c^* \geq 0$ and take $\mathbf{x}^* \coloneqq \operatorname{argmin}_{\mathbf{x}\in\mathbb{R}^{[d]}}\{l(\mathbf{x}) \mid G(\mathbf{x}) \leq c^*\}$ to be a target solution; our guarantees will be parametrized with c and c^* as shown later. We also define $k^* \coloneqq \|\mathbf{x}^*\|_0$.

3 COST-BENEFIT GREEDY

We consider a greedy-style algorithm for problem (1). As in (Krause and Cevher, 2010; Bogunovic et al., 2018), we define a set function as follows:

$$F(\mathsf{S}) \coloneqq l(0) - \min_{\mathrm{supp}(\mathbf{x})\subseteq\mathsf{S}} l(\mathbf{x}), \tag{3}$$

where $\mathsf{S} \subseteq [d]$. Note that $F(\cdot)$ is monotone and satisfies $F(\emptyset) = 0$. We assume that $F(\mathsf{S})$ can easily be evaluated for any $\mathsf{S} \in \mathcal{F}$, which is true in many cases: If $l(\cdot)$ is a quadratic loss function, $F(\mathsf{S})$ can be obtained by computing a pseudo-inverse matrix. Given more general objective functions, we can use iterative methods (e.g., (Shalev-Shwartz and Zhang, 2016)) to compute the minimum in (3). With $F(\cdot)$ thus defined, we reformulate problem (1) as the following set-function maximization problem:

$$\underset{\mathsf{S}\subseteq[d]}{\text{maximize}} \;\; F(\mathsf{S}) \quad \text{subject to } G(\mathsf{S}) \leq c. \tag{4}$$

Thanks to the result of (Elenberg et al., 2018), the above problem can be seen as a constrained weakly submodular function maximization problem. Formally, $F(\cdot)$ has the following property:

Proposition 1 (Elenberg et al. (2018))**.** *For any* $\mathsf{U} \subseteq [d]$ *and* $k \in \mathbb{Z}_{>0}$*, submodularity ratio* $\gamma_{\mathsf{U},k}$ *of* $F(\cdot)$ *is bounded with RSC and RSM constants of* $l(\cdot)$ *as*

$$\gamma_{\mathsf{U},k} \geq \frac{\mu_{|\mathsf{U}|+k}}{\nu_{|\mathsf{U}|+1,1}} \geq \frac{\mu_{|\mathsf{U}|+k}}{\nu_{|\mathsf{U}|+k}}.$$

Algorithm 1 Cost-benefit greedy

```
$\mathsf{U} \leftarrow [d]$, $\mathsf{S} \leftarrow \emptyset$
while $\mathsf{U} \neq \emptyset$ do
    $j \leftarrow \operatorname{argmax}_{j'\in\mathsf{U}} \frac{F(j'|\mathsf{S})}{G(j'|\mathsf{S})}$
    if $G(\mathsf{S}\cup\{j\}) \leq c$ then
        $\mathsf{S} \leftarrow \mathsf{S}\cup\{j\}$
    $\mathsf{U} \leftarrow \mathsf{U}\backslash\{j\}$
return $\mathsf{S}$
```

Algorithm 1 presents the details of the cost-benefit greedy algorithm (CBG) for problem (4), which is a set-function constraint version of the greedy algorithm studied in (Sviridenko, 2004; Leskovec et al., 2007). A similar algorithm, called StructOMP, was also studied in (Huang et al., 2011) for the case where $l(\cdot)$ is quadratic and the constraint is defined using subadditive coding complexity. Unlike this result, our result accepts any $l(\cdot)$ with RSC/RSM and monotone cost functions with positive restricted inverse curvature.

Theorem 1. *Let* S *be the output of CBG and* S^* *be any subset that satisfies* $G(\mathsf{S}^*) \leq c^*$ *and* $|\mathsf{S}^*| = k^*$*. If* $F(\cdot)$ *has submodularity ratio* γ_{S,k^*}*,* $G(\cdot)$ *has superadditivity ratio* β_{k^*} *and restricted inverse curvature* θ*, and* $\min\{c, c^*\} \geq \rho$ *holds, then we have*

$$F(\mathsf{S}) \geq \left(1 - \exp\left(-\theta\beta_{k^*}\gamma_{\mathsf{S},k^*} \cdot \frac{c-\rho}{c^*}\right)\right) F(\mathsf{S}^*).$$

By using definition (3), Proposition 1, and Theorem 1, we obtain the following guarantee of CBG for problem (1):

Corollary 1.a. *Let* $\mathbf{x} \coloneqq \operatorname{argmin}_{\mathrm{supp}(\mathbf{x}')\subseteq\mathsf{S}} l(\mathbf{x}')$ *and* $k \coloneqq |\mathsf{S}|$*. If* $l(\cdot)$ *is* μ_{k+k^*}*-RSC and* $\nu_{k+1,1}$*-RSM, then*

$$l(\mathbf{x}) \leq l(\mathbf{x}^*) + \exp\left(-\theta\beta_{k^*}\frac{\mu_{k+k^*}}{\nu_{k+1,1}} \cdot \frac{c-\rho}{c^*}\right)(l(0) - l(\mathbf{x}^*))$$

holds. In particular, for any $\epsilon > 0$*, if we have*

$$c \geq \frac{c^*}{\theta\beta_{k^*}} \cdot \frac{\nu_{k+1,1}}{\mu_{k+k^*}} \log\frac{l(0) - l(\mathbf{x}^*)}{\epsilon} + \rho,$$

then $l(\mathbf{x}) \leq l(\mathbf{x}^*) + \epsilon$ *holds.*

As in (Shalev-Shwartz et al., 2010), in the case of ℓ_0-constrained minimization, whose constraint is given by $\|\mathbf{x}\|_0 \leq k$, the greedy algorithm is known to achieve an ϵ-error by setting

$$k \geq \|\mathbf{x}^*\|_0 \cdot \frac{\nu}{\mu}\log\frac{l(0) - l(\mathbf{x}^*)}{\epsilon},$$

where $\mu \coloneqq \mu_{\mathbb{R}^{[d]}\times\mathbb{R}^{[d]}}$ and $\nu \coloneqq \nu_{\mathbb{R}^{[d]}\times\mathbb{R}^{[d]}}$ are SC and SM constants, respectively. Therefore, our result can be seen as a set-function-constraint version of the existing result for ℓ_0-constrained minimization, where the difficulty of dealing with general cost functions is represented using parameters: θ and β_{k^*}.

Algorithm 2 IHT with CBG projection

```
1: Initialize $\mathbf{x}_0 \in \mathbb{R}^{[d]}$
2: for $t = 0, 1, \dots, T-1$ do
3:     $\mathbf{g}_t \leftarrow \mathbf{x}_t - \eta \nabla l(\mathbf{x}_t)$
4:     $\mathbf{x}_{t+1} \leftarrow \mathcal{P}_c(\mathbf{g}_t)$
5: return $\mathbf{x}_T$
```

4 IHT WITH CBG PROJECTION

We consider applying an IHT-style method (Algorithm 2) to problem (1). Similarly to the standard IHT (Jain et al., 2014), the algorithm iteratively updates a solution via gradient descent and projection onto the feasible region. However, since the constraint is given by a set function, $G(\cdot)$, we need a projection step that works well with $G(\cdot)$. Here, we use the CBG algorithm to perform the projection step, which is denoted by $\mathcal{P}_c(\mathbf{g}_t)$. More precisely, we execute Algorithm 1 with objective function $F(\mathsf{S}) = \|(\mathbf{g}_t)_\mathsf{S}\|_2^2$, cost function $G(\mathsf{S})$, and budget value c; we thus obtain solution S. We then set $(\mathbf{x}_{t+1})_j$ to $(\mathbf{g}_t)_j$ if $j \in \mathsf{S}$ and 0 otherwise. This projection step is similar to that of (Jain et al., 2016), but there is a technical difference between them as explained in Section 1.2. Thanks to Theorem 1, we can evaluate the performance of $\mathcal{P}_c(\cdot)$, from which we can obtain the following convergence guarantee of IHT with CBG projection:

Theorem 2. *Let* $k := \max_{t:0\le t\le T} \|\mathbf{x}_t\|_0$ *and* $\omega := \max_{t:0\le t\le T} \|\mathbf{g}_t\|_2$. *Assume that* $l(\cdot)$ *is continuously twice differentiable,* μ_{2k+k^*}*-RSC, and* ν_{2k+k^*}*-RSM, and that* $G(\cdot)$ *has superadditivity ratio* β_{k^*} *and restricted inverse curvature* θ. *Set* $\eta = \frac{1}{\nu_{2k+k^*}}$. *If* $c^* \ge \rho$ *and* $c \ge \frac{4c^*}{\theta\beta_{k^*}} \left(\frac{\nu_{2k+k^*}}{\mu_{2k+k^*}}\right)^2 + \frac{2c^*}{\theta\beta_{k^*}} \log\left(\frac{\omega}{2\epsilon}\right) + 2\rho$ *hold, then we have*

$$\|\mathbf{x}_{t+1} - \mathbf{x}^*\|_2 \\ \le \left(1 - \frac{1}{2} \cdot \frac{\mu_{2k+k^*}}{\nu_{2k+k^*}}\right) \|\mathbf{x}_t - \mathbf{x}^*\|_2 + \zeta + \frac{\mu_{2k+k^*}}{\nu_{2k+k^*}} \cdot \epsilon,$$

where $\zeta := \frac{1}{\nu_{2k+k^*}} \left(1 + \frac{1}{2} \cdot \frac{\mu_{2k+k^*}}{\nu_{2k+k^*}}\right) \max_{\mathsf{S}\in\mathcal{F}} \|\nabla l(\mathbf{x}^*)_\mathsf{S}\|_2$. *Specifically, after* $T \ge 2 \cdot \frac{\nu_{2k+k^*}}{\mu_{2k+k^*}} \log \frac{\|\mathbf{x}_0 - \mathbf{x}^*\|_2}{\epsilon}$ *steps, we have* $\|\mathbf{x}_T - \mathbf{x}^*\|_2 \le 3\epsilon + 2\zeta \cdot \frac{\nu_{2k+k^*}}{\mu_{2k+k^*}}$.

Namely, given appropriate c, we can recover $\mathbf{x}^*$ with up to $O(\zeta \cdot \frac{\nu_{2k+k^*}}{\mu_{2k+k^*}})$ error; in particular, $\zeta = 0$ holds if $\mathbf{x}_{\min} := \operatorname{argmin}_{\mathbf{x}\in\mathbb{R}^{[d]}} l(\mathbf{x})$ satisfies $G(\operatorname{supp}(\mathbf{x}_{\min})) \le c^*$. As in (Jain et al., 2014), IHT achieves an ϵ-error for ℓ_0-constrained minimization if $k \ge \Omega\left(\left(\frac{\nu_{2k+k^*}}{\mu_{2k+k^*}}\right)^2 \|\mathbf{x}^*\|_0\right)$ and $T \ge \Omega\left(\frac{\nu_{2k+k^*}}{\mu_{2k+k^*}} \log\left(\frac{l(\mathbf{x}_0)}{\epsilon}\right)\right)$. Therefore, our result can again be seen as a set-function-constraint version of the existing result for ℓ_0-constrained minimization.

Below we detail additional techniques that can improve the empirical performance of the algorithm.

Backtracking for Step-size Computation In experiments, we use the following standard backtracking to compute η in Step 3. Let η_t denote the step size used in the t-th iteration. We first set $\eta_t \leftarrow 2\eta_{t-1}$ if $t \ge 2$ and $\eta_t = 1$ if $t = 1$. If $l(\mathbf{g}_t) > l(\mathbf{x}_t)$ occurs with the current step size, η_t, we then set $\eta_t \leftarrow \eta_t/2$ and recompute $\mathbf{g}_t$; we repeat this until we get $l(\mathbf{g}_t) \le l(\mathbf{x}_t)$. We thus guarantee that the objective value always decreases via gradient descent in Step 3.

Full Correction The following *full-correction* step can be incorporated into Algorithm 2: After obtaining $\mathbf{x}_{t+1}$ in Step 4, we set $\mathbf{x}_{t+1} \leftarrow \operatorname{argmin}_{\mathbf{x}\in\mathbb{R}^{[d]}}\{l(\mathbf{x}) \mid \operatorname{supp}(\mathbf{x}) \subseteq \operatorname{supp}(\mathbf{x}_{t+1})\}$. This technique is often used to speed up IHT, and so we used it in our experiments.

5 APPLICATIONS

An advantage of the above CBG and IHT is that it can be applied to various problems of form (1). Below we list examples of cost functions, $G(\cdot)$, and we show that their superadditivity ratio, β_k, and restricted inverse curvature, θ, can be bounded.

Non-subadditive Costs Most existing works on greedy algorithms that deal with set-function-based constraints assume that the set functions have subadditivity (Huang et al., 2011) or submodularity (Iyer and Bilmes, 2013). However, those properties are not always satisfied in realistic situations. To be concrete, let $\mathsf{B}_1, \mathsf{B}_2 \subseteq [d]$ be two fixed subsets, which may overlap. We consider a sparse regression problem, where $[d]$ represents the full set of features; we suppose that B_1 and B_2 represent two groups of expensive features and that to use any $j_1 \in \mathsf{B}_1$ and $j_2 \in \mathsf{B}_2$ simultaneously is very expensive. To force the resulting solutions to be sparse and cheap, it is natural to use $G(\mathsf{S}) = |\mathsf{S}| + C(\mathsf{S})$ as a cost function, where $C(\mathsf{S})$ is defined as

$$C(\mathsf{S}) := \begin{cases} a & \text{if either } \mathsf{S}\cap\mathsf{B}_1 \ne \emptyset \text{ or } \mathsf{S}\cap\mathsf{B}_2 \ne \emptyset, \\ b & \text{if } \mathsf{S}\cap\mathsf{B}_1 \ne \emptyset \text{ and } \mathsf{S}\cap\mathsf{B}_2 \ne \emptyset, \\ 0 & \text{otherwise} \end{cases}$$

with some $0 \le a \le b$. The function is monotone, but it is neither submodular nor subadditive; if $\mathsf{S}_1 \subseteq \mathsf{B}_1 \backslash \mathsf{B}_2$, $\mathsf{S}_2 \subseteq \mathsf{B}_2 \backslash \mathsf{B}_1$, and $b > 2a$, then we have $C(\mathsf{S}_1) + C(\mathsf{S}_2) < C(\mathsf{S}_1 \cup \mathsf{S}_2) \le C(\mathsf{S}_1 \cup \mathsf{S}_2) + C(\mathsf{S}_1 \cap \mathsf{S}_2)$. On the other hand, since we have $\frac{|\mathsf{S}| + C(\mathsf{S})}{|\mathsf{S}| + \sum_{j\in\mathsf{S}} C(j)} \ge \frac{1}{1 + \frac{1}{|\mathsf{S}|}\sum_{j\in\mathsf{S}} C(j)} \ge \frac{1}{1+b}$ for any $\mathsf{S} \subseteq [d]$, superadditivity ratio β_k is bounded from below by $\frac{1}{1+b}$. Furthermore, by considering each case separately, we can confirm that restricted inverse curvature θ of G is bounded from below by $\min\{1, \frac{1+a}{1+(b-a)}\}$.

It is not difficult to extend the above discussion to the case where there are more groups of expensive features. In the experiment (Section 6.4), we show that such a cost function naturally arises when predicting the status of patients from data collected via some tests.

Contiguous Sparsity Suppose that the entries of $\mathbf{x} \in \mathbb{R}^{[d]}$ are arranged on a 1D line. We aim to obtain sparse $\mathbf{x}$ such that $\mathrm{supp}(\mathbf{x})$ has a small number of intervals (see, the right figure in Figure 3). As in (Bach, 2010), the following submodular function is often used to obtain such a solution: $G(\mathsf{S}) := \xi|\mathsf{S}| + \mathrm{NI}(\mathsf{S})$, where $\xi \geq 0$ is a hyper-parameter and $\mathrm{NI}(\mathsf{S})$ is the number of intervals formed by S (e.g., $\mathrm{NI}(\{1, 2, 4, 5\}) = 2$). More generally, given a graph and vertex subset S, we can define $\mathrm{NI}(\mathsf{S})$ as the number of connected components induced by S. The resulting constraint forces solutions to have a small number of non-zeros and connected components, which can be seen as a variant of the constraint considered in (Hegde et al., 2015). Let $\deg \geq 0$ be the largest degree in the graph, and assume that ξ is set to satisfy $\xi \geq \deg - 1$, which makes function G monotone. Since G is submodular, we have $\theta \geq 1 - \check{\alpha} = 1$. Furthermore, since $\frac{\xi|\mathsf{S}|+\mathrm{NI}(\mathsf{S})}{\xi|\mathsf{S}|+\sum_{j\in\mathsf{S}}\mathrm{NI}(j)} \geq \frac{\xi|\mathsf{S}|+1}{\xi|\mathsf{S}|+\sum_{j\in\mathsf{S}}\mathrm{NI}(j)} \geq \frac{\xi+k^{-1}}{\xi+1}$ holds for any S of size at most k, we have $\beta_k \geq \frac{\xi+k^{-1}}{\xi+1}$.

Submodular Costs with Bounded Curvature As in (Sharma et al., 2015; Maehara et al., 2017), various monotone submodular functions have bounded curvature $\alpha \in [0, 1]$. As shown in (Bogunovic et al., 2018), superadditivity ratio β_k is bounded from below by $1 - \alpha$, and restricted inverse curvature θ of submodular functions is equal to 1. In particular, if $G(\cdot)$ is modular, which corresponds to the case of a knapsack constraint, we have $\theta = \beta_k = 1$. Therefore, our analysis can be applied to any monotone submodular cost function whose curvature α is bounded from above.

Concave Functions of Non-negative Weights Let $h : [0, \infty) \to [0, \infty)$ be a non-decreasing concave function such that $h(0) = 0$, and suppose that each $j \in [d]$ is associated with non-negative weight $w_j \geq 0$. Then, $G(\mathsf{S}) := h(\sum_{j\in\mathsf{S}} w_j)$ is known to be monotone and submodular (Bach, 2013); hence $\theta = 1$. Of particular interest, letting $p \in [1, \infty)$ and regarding $w_j^p \geq 0$ as a weight value, the p-norm function defined as $G(\mathsf{S}) = \|\mathbf{w}_\mathsf{S}\|_p := (\sum_{j\in\mathsf{S}} w_j^p)^{1/p}$ is monotone and submodular, where $\mathbf{w} := (w_1, \dots, w_d)^\top$. Thanks to Hölder's inequality, we have $\sum_{j\in\mathsf{S}} w_j \leq \|\mathbf{w}_\mathsf{S}\|_p |\mathsf{S}|^{\frac{p-1}{p}}$, which means $\beta_k \geq k^{-\frac{p-1}{p}}$.

Spectral Functions of Submatrix Given matrix $\mathbf{A} \in \mathbb{R}^{[n]\times[d]}$, monotone submodular functions, $G(\mathsf{S})$, defined with submatrix $\mathbf{A}_\mathsf{S}$ are used in many scenarios. One such example is the trace norm function, $G(\mathsf{S}) = \sqrt{\mathrm{tr}(\mathbf{A}_\mathsf{S}^\top \mathbf{A}_\mathsf{S})}$, which is a composition of $h(x) = \sqrt{x}$ and $\sum_{j\in\mathsf{S}} \|\mathbf{A}_j\|_2^2$; hence its superadditivity ratio and restricted inverse curvature are bounded as above. In sparse Bayesian learning, $G(\mathsf{S}) = \log(\det(\xi\mathbf{I}_{\mathsf{S},\mathsf{S}} + \mathbf{A}_\mathsf{S}^\top \mathbf{A}_\mathsf{S}))$ is often used (Wipf and Nagarajan, 2009; Bach, 2010), which is monotone if $\xi \geq 1$. We define $\mathbf{X} := \xi\mathbf{I} + \mathbf{A}^\top\mathbf{A}$ and $x_{\max} := \max_{j\in[d]} \mathbf{X}_{j,j}$. Let $\lambda_{\min}$ be the smallest eigenvalue of $\mathbf{X}$. If $\lambda_{\min} > 1$, then $\beta_k = \min_{|\mathsf{S}|\leq k} \frac{\log(\det(\mathbf{X}_{\mathsf{S},\mathsf{S}}))}{\sum_{j\in\mathsf{S}} \log \mathbf{X}_{j,j}} \geq \frac{\log \lambda_{\min}}{\log x_{\max}}$ holds. Furthermore, we have $\theta = 1$ from the submodularity.

6 EXPERIMENTS

We conduct experiments to study behavior of CBG and IHT. In Sections 6.1, 6.2, and 6.3, we use synthetic instances of regression with contiguous sparsity to elucidate typical behavior of CBG and IHT with various settings. In Section 6.4, we apply CBG and IHT to instances with real-world data.

In Sections 6.1–6.3, we use fused lasso (Tibshirani et al., 2005) as a baseline method, which is a convex-relaxation method to deal with contiguous sparsity. With fused lasso, a solution is obtained by solving $\min_{\mathbf{x}\in\mathbb{R}^{[d]}} l(\mathbf{x}) + \lambda_1\|\mathbf{x}\|_1 + \lambda_2 \sum_{i=2}^{d} |\mathbf{x}_i - \mathbf{x}_{i-1}|$. We used the efficient fused lasso algorithm (EFLA) (Liu et al., 2010) to solve the problem. We applied fused lasso with 16 pairs of parameters, $(\lambda_1, \lambda_2) \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}^2$, to the instances, and we found that the overall performance is good with $(\lambda_1, \lambda_2) = (10^{-2}, 10^{-3})$; hence this pair is used in the following experiments.

All our experiments were conducted on a 64-bit macOS (High Sierra) machine with 3.3GHz Intel Core i7 CPUs and 16 GB RAM. With IHT and fused lasso, we continued the iteration until the objective-value improvement, $l(\mathbf{x}_t) - l(\mathbf{x}_{t+1})$, became smaller than 10^{-8}.

Summary of Results Since the experiments are very extensive, we here summarize the results: IHT works well with well-conditioned instances (Section 6.1), while CBG is effective for ill-conditioned instances in terms of solution quality (Section 6.2). Both CBG and IHT can work well even if the objective functions are non-convex, while fused lasso cannot (Section 6.3). Section 6.4 demonstrates that CBG and IHT can work with a non-subadditive cost function. We also see that CBG and IHT have complementary natures, which emphasizes the importance of studying both of them.

6.1 Well-conditioned Instances

We consider a regression model with contiguous sparsity on a 1D line; given a sample of size n, we estimate

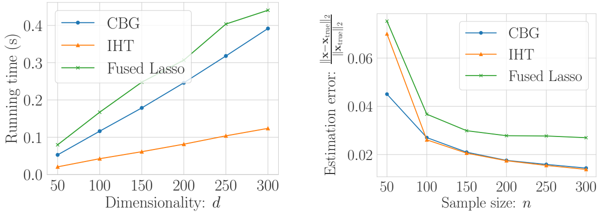

Figure 1: Results of well-conditioned instances. The left and right figures show the running times and estimation errors of the three methods, respectively. Each value is calculated by taking an average over 100 instances.

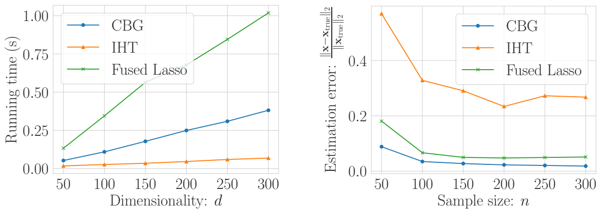

Figure 2: Comparisons of running times and estimation errors for ill-conditioned instances. Each value is calculated by taking an average over 100 instances.

$\mathbf{x} \in \mathbb{R}^{[d]}$ that has a small number of non-zeros and intervals. Given design matrix $\mathbf{A} \in \mathbb{R}^{[n]\times[d]}$ and observation vector $\mathbf{y} \in \mathbb{R}^{[n]}$, we use a quadratic loss function: $l(\mathbf{x}) := \frac{1}{2n}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$. We randomly generated 100 instances as follows. We chose $\mathsf{S}_{\text{true}} \subseteq [d]$ of size k, which corresponds to non-zeros of the true solution, $\mathbf{x}_{\text{true}}$, so as to have only two intervals. Each non-zero entry of $\mathbf{x}_{\text{true}}$ was chosen uniformly at random from $[-1, 1]$. As a cost function, we used $G(\mathsf{S}) = 2|\mathsf{S}| + \mathrm{NI}(\mathsf{S})$. In this well-conditioned setting, we drew each entry of $\mathbf{A} \in \mathbb{R}^{[n]\times[d]}$ from the standard normal distribution, which we denote by $\mathcal{N}$, and we set $\mathbf{y}_{\text{true}} = \mathbf{A}\mathbf{x}_{\text{true}}$. We then set $\mathbf{y} = \mathbf{y}_{\text{true}} + 0.1\mathbf{u}$, where each entry of $\mathbf{u} \in \mathbb{R}^{[n]}$ was drawn from $\mathcal{N}$. We thus obtained $\mathbf{A}$ and $\mathbf{y}$. Throughout this section, we set $k = 10$, and we investigate how the sample size, n, and the dimensionality, d, affect the performance of the algorithms. The budget value was set as $c = 1.25(k/d) \times G([d])$. We evaluate the three methods (CBG, IHT, and fused lasso) by running time and estimation error, $\|\mathbf{x} - \mathbf{x}_{\text{true}}\|_2/\|\mathbf{x}_{\text{true}}\|_2$, where $\mathbf{x}$ is an output of the algorithms.

Results The results are presented in Figure 1. The left figure shows the running times of the three methods, where the dimensionality varies as $d = 50, 100, \ldots, 300$ and we set $n = \lfloor k \log d \rfloor$. We see that IHT is far faster than the other methods. The right figure indicates estimation errors of each method; here, we set $d = 100$ and the sample size varies as $n = 50, 100, \ldots, 300$. Except for the case of $n = 50$, IHT achieves the smallest estimation error. These results suggest that IHT is the overall winner when instances are well-conditioned and sufficiently large samples are available.

6.2 Ill-conditioned Instances

The problem setting used in this section is almost the same as that of the previous section. The only difference is the construction of design matrix $\mathbf{A} \in \mathbb{R}^{[n]\times[d]}$: The matrix consists of $\lfloor n/2 \rfloor$ rows that are drawn from a heavily correlated d-dimensional normal distribution, whose correlation coefficient is set to 0.8, and $\lceil n/2 \rceil$ rows whose elements are drawn from $\mathcal{N}$. Matrix $\mathbf{A}$ thus obtained has a larger condition number than the previous one. We thus generated 100 ill-conditioned random instances.

Results The results are shown in Figure 2. As with the well-conditioned case, IHT is the fastest. Compared to the well-conditioned case, fused lasso is much slower than CBG. This is because the speed of fused lasso, a convex-optimization approach, is more negatively impacted by the ill-condition than is CBG. More precisely, the speeds of fused lasso and CBG generally depend on the condition number, $\kappa := \nu/\mu$, and restricted condition number, κ_Ω, respectively; with ill-conditioned instances, κ tends to become much larger than κ_Ω, and thus fused lasso slows down much more. Regarding estimation errors, unlike the well-conditioned case, the performance of IHT is the poorest, and CBG achieves the smallest errors. This is consistent with our theoretical results: To obtain the guarantees, IHT and CBG require that the budget value, c, is larger than $\Omega(\kappa_\Omega^2)$ and $\Omega(\kappa_\Omega)$, respectively, which implies IHT is more vulnerable to the ill-condition than CBG.

6.3 Non-convex Objective Functions

We again consider regression problems with contiguous sparsity. We set $d = 100$ and $k = 10$. As in the previous sections, $\mathbf{x}_{\text{true}} \in \mathbb{R}^{[d]}$ has k non-zeros, which form only two intervals as in Figure 3; we here set all non-zeros of $\mathbf{x}_{\text{true}}$ to 1. To create a non-convex objective function, we first randomly generated a graph Laplacian matrix, $\mathbf{L} \in \mathbb{R}^{[d]\times[d]}$, whose smallest eigenvalue is always equal to 0. We then set $l(\mathbf{x}) = \frac{1}{2d}(\mathbf{x} - \mathbf{x}_{\text{true}})^\top \mathbf{H}(\mathbf{x} - \mathbf{x}_{\text{true}})$, where $\mathbf{H} := \mathbf{L} - \omega\mathbf{I}$; the smallest eigenvalue of the Hesse matrix, $\mathbf{H}/d$, is $-\omega/d$, and so $\omega \geq 0$ can be seen as a parameter that controls the non-convexity of $l(\cdot)$. On the other hand, as long as ω is small, the objective function is RSC/RSM over the feasible region. We observed that $l(\cdot)$ became non-convex over the feasible region when $\omega \geq 1$; in this case the optimal value can become arbitrarily small. Therefore, we consider $l(\cdot)$ with $\omega = 0, 0.2, \ldots, 0.8$.

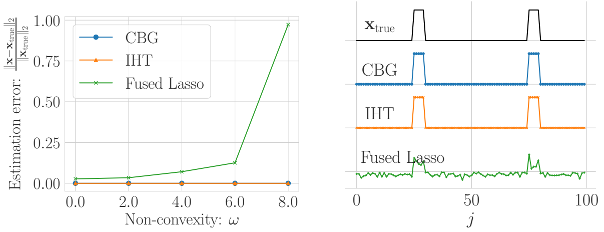

Figure 3: Results of an instance with a non-convex objective function. The left figure shows estimation errors of each method. The right figure illustrates $\mathbf{x}_{\mathrm{true}}$ and solutions of the three methods for $\omega = 0.8$.

Results Figure 3 shows the results. As in the left figure, CBG and IHT achieve small estimation errors for every ω, while those of fused lasso increase with ω. The right figure illustrates $\mathbf{x}_{\mathrm{true}}$ and the solutions obtained with the three methods for $\omega = 0.8$. We see that CBG and IHT successfully recovered $\mathbf{x}_{\mathrm{true}}$, while fused lasso failed to recover $\mathbf{x}_{\mathrm{true}}$. Note that fused lasso is expected to perform well in this setting since the true solution, $\mathbf{x}_{\mathrm{true}}$, has only four pairs of adjacent entries, $(\mathbf{x}_{i-1}, \mathbf{x}_i)$, satisfying $\mathbf{x}_{i-1} \neq \mathbf{x}_i$; in such cases, the fusion penalty, $\lambda_2 \sum_{i=2}^{d} |\mathbf{x}_i - \mathbf{x}_{i-1}|$, typically works well. Therefore, the performance decline of fused lasso is purely due to the increase in non-convexity. These results suggest that CBG and IHT are advantageous when the objective functions are non-convex.

6.4 Real-world Instances

We consider sparse regression instances with real-world data. As in Sections 6.1 and 6.2, the objective function is defined as $l(\mathbf{x}) \coloneqq \frac{1}{2n}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$. Observation vector $\mathbf{y}$ and design matrix $\mathbf{A}$ were obtained from Diabetes data, which is available as a scikit-learn dataset. The original data has 10 features: age, sex, bmi, average blood pressure (abp), and six attributes (S1–S6) whose values are obtained via a blood test. As in (Bertsimas et al., 2016), we consider interaction of the original features; consequently, we have $d = 10 + \binom{10}{2} = 55$ features. The dataset has a sample of size $n = 442$, and we split them into training data ($n = 200$) and test data ($n = 242$). With the data thus obtained, we perform regression to obtain a sparse linear model, which we use to predict the status of patients. Here, we consider a cost function that forces output solutions to be sparse and burdenless; to obtain the values of abp and S1–S6, we need to conduct a blood pressure test and blood test, respectively, which are burdensome for a patient, and to do both of the tests imposes much of a burden. Thus, to avoid using features that requires burdensome tests, we use cost function $G(\mathsf{S}) \coloneqq |\mathsf{S}| + C(\mathsf{S})$, where $C : 2^{[d]} \to \mathbb{R}$ is defined as follows: Let $\mathsf{B}_1, \mathsf{B}_2 \subseteq [d]$ be two subsets of burdensome features that are associated with abp and S1–S6, respectively; for example, an interaction feature that consists of age and abp is included in B_1. Given budget value c, we define

$$C(\mathsf{S}) \coloneqq \begin{cases} 0.1c & \text{if either } \mathsf{S} \cap \mathsf{B}_1 \neq \emptyset \text{ or } \mathsf{S} \cap \mathsf{B}_2 \neq \emptyset, \\ 0.3c & \text{if } \mathsf{S} \cap \mathsf{B}_1 \neq \emptyset \text{ and } \mathsf{S} \cap \mathsf{B}_2 \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases}$$

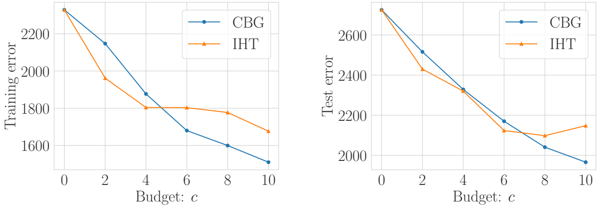

Figure 4: Training and test errors of CBG and IHT with various budget values.

We consider various budget values $c = 0, 2, \ldots, 10$. To the best of our knowledge, no methods other than CBG and IHT can work with the above cost function, and thus we apply the two methods to the problem and evaluate their performances.

Results Figure 4 shows the results. When c was small (about $c \leq 4$), IHT outperformed CBG. Actually, IHT successfully avoided choosing burdensome features, while CBG had trouble handling the constraints with small c; in fact, all solutions $\mathbf{x}$ obtained with CBG satisfied $\mathrm{supp}(\mathbf{x}) \cap \mathsf{B}_1 \neq \emptyset$ and $\mathrm{supp}(\mathbf{x}) \cap \mathsf{B}_2 \neq \emptyset$ for $c \geq 2$, which leads to excessive cost values. When c was large, however, IHT behaved somewhat conservatively; even with sufficiently large c, the solutions of IHT did not use features included in B_1. In contrast, CBG achieved small training errors by aggressively choosing features that are burdensome but effective for reducing objective values. These results suggest that CBG and IHT can exhibit complementary natures; i.e., IHT is better than CBG when c is small, and the opposite is true when c is large. In practice, it can be beneficial to apply both CBG and IHT to a given instance and use the solution with the smaller training error.

7 CONCLUSION

We proved theoretical guarantees of CBG and IHT for non-convex optimization problems with monotone cost function constraints. We provided examples of monotone cost functions and showed that their super-additivity ratio and restricted inverse curvature can be bounded. Experiments showed typical behavior and advantages of the considered algorithms.

Acknowledgements

The author is grateful to Takanori Maehara, Masaaki Nishino, and Tasuku Soma for valuable comments.

References

Bach, F. (2010). Structured sparsity-inducing norms through submodular functions. In *Advances in Neural Information Processing Systems 23*, pages 118–126. Curran Associates, Inc.

Bach, F. (2013). Learning with submodular functions: A convex optimization perspective. *Foundations and Trends® in Machine Learning*, 6(2-3):145–373.

Barber, R. F. and Ha, W. (2018). Gradient descent with non-convex constraints: local concavity determines convergence. *Inf. Inference*, 7(4):755–806.

Bertsimas, D., King, A., and Mazumder, R. (2016). Best subset selection via a modern optimization lens. *Ann. Statist.*, 44(2):813–852.

Bian, A. A., Buhmann, J. M., Krause, A., and Tschiatschek, S. (2017). Guarantees for greedy maximization of non-submodular functions with applications. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 498–507. PMLR.

Blumensath, T. and Davies, M. E. (2009). Iterative hard thresholding for compressed sensing. *Appl. Comput. Harmon. Anal.*, 27(3):265–274.

Bogunovic, I., Zhao, J., and Cevher, V. (2018). Robust maximization of non-submodular objectives. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 890–899. PMLR.

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

Chen, L., Feldman, M., and Karbasi, A. (2018). Weakly submodular maximization beyond cardinality constraints: Does randomization help greedy? In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 804–813. PMLR.

Chen, X., Lin, Q., Kim, S., Carbonell, J. G., and Xing, E. P. (2012). Smoothing proximal gradient method for general structured sparse regression. *Ann. Appl. Stat.*, 6(2):719–752.

Das, A. and Kempe, D. (2011). Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In *Proceedings of the 28th International Conference on Machine Learning*, pages 1057–1064. ACM.

Elenberg, E. R., Khanna, R., Dimakis, A. G., and Negahban, S. (2018). Restricted strong convexity implies weak submodularity. *Ann. Statist.*, 46(6B):3539–3568.

Foucart, S. (2011). Hard thresholding pursuit: An algorithm for compressive sensing. *SIAM J. Optim.*, 49(6):2543–2563.

Hegde, C., Indyk, P., and Schmidt, L. (2015). A nearly-linear time framework for graph-structured sparsity. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 928–937. PMLR.

Huang, J., Zhang, T., and Metaxas, D. (2011). Learning with structured sparsity. *J. Mach. Learn. Res.*, 12:3371–3412.

Iyer, R. K. and Bilmes, J. A. (2013). Submodular optimization with submodular cover and submodular knapsack constraints. In *Advances in Neural Information Processing Systems 26*, pages 2436–2444. Curran Associates, Inc.

Jain, P. and Kar, P. (2017). Non-convex optimization for machine learning. *Foundations and Trends® in Machine Learning*, 10(3-4):142–336.

Jain, P., Rao, N., and Dhillon, I. S. (2016). Structured sparse regression via greedy hard thresholding. In *Advances in Neural Information Processing Systems 29*, pages 1516–1524. Curran Associates, Inc.

Jain, P., Tewari, A., and Kar, P. (2014). On iterative hard thresholding methods for high-dimensional M-estimation. In *Advances in Neural Information Processing Systems 27*, pages 685–693. Curran Associates, Inc.

Khanna, R. and Kyrillidis, A. (2018). IHT dies hard: Provable accelerated iterative hard thresholding. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, volume 84, pages 188–198. PMLR.

Krause, A. and Cevher, V. (2010). Submodular dictionary selection for sparse representation. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 567–574. Omnipress.

Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., and Glance, N. (2007). Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 420–429. ACM.

Liberty, E. and Sviridenko, M. (2017). Greedy minimization of weakly supermodular set functions. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 81, pages 19:1–19:11. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

Liu, J., Yuan, L., and Ye, J. (2010). An efficient algorithm for a class of fused lasso problems. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 323–332. ACM.

Maehara, T., Kawase, Y., Sumita, H., Tono, K., and Kawarabayashi, K. (2017). Optimal pricing for submodular valuations with bounded curvature. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*.

Needell, D. and Tropp, J. A. (2009). CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Appl. Comput. Harmon. Anal.*, 26(3):301 – 321.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions-I. *Math. Program.*, 14(1):265–294.

Pati, Y. C., Rezaiifar, R., and Krishnaprasad, P. S. (1993). Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Proceedings of the 27th Asilomar Conference on Signals, Systems and Computers*, pages 40–44 vol.1.

Qian, C., Yu, Y., and Tang, K. (2018). Approximation guarantees of stochastic greedy algorithms for subset selection. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 1478–1484. International Joint Conferences on Artificial Intelligence Organization.

Shalev-Shwartz, S., Srebro, N., and Zhang, T. (2010). Trading accuracy for sparsity in optimization problems with sparsity constraints. *SIAM J. Optim.*, 20(6):2807–2832.

Shalev-Shwartz, S. and Zhang, T. (2016). Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Math. Program.*, 155(1):105–145.

Sharma, D., Kapoor, A., and Deshpande, A. (2015). On greedy maximization of entropy. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1330–1338. PMLR.

Sviridenko, M. (2004). A note on maximizing a submodular set function subject to a knapsack constraint. *Oper. Res. Lett.*, 32(1):41–43.

Tewari, A., Ravikumar, P. K., and Dhillon, I. S. (2011). Greedy algorithms for structurally constrained high dimensional problems. In *Advances in Neural Information Processing Systems 24*, pages 882–890. Curran Associates, Inc.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and smoothness via the fused lasso. *J. R. Stat. Soc. Ser. B. Stat. Methodol.*, 67(1):91–108.

Wipf, D. P. and Nagarajan, S. S. (2009). Sparse estimation using general likelihoods and non-factorial priors. In *Advances in Neural Information Processing Systems 22*, pages 2071–2079. Curran Associates, Inc.

Yuan, X., Li, P., and Zhang, T. (2016). Exact recovery of hard thresholding pursuit. In *Advances in Neural Information Processing Systems 29*, pages 3558–3566. Curran Associates, Inc.

Zhang, H. and Vorobeychik, Y. (2016). Submodular optimization with routing constraints. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 819–825. AAAI Press.

Zhou, B. and Chen, F. (2016). Graph-structured sparse optimization for connected subgraph detection. In *Proceedings of the 16th International Conference on Data Mining*, pages 709–718. IEEE.