ON SPARSE MIRROR DESCENT

BY

SHOVIK GUHA

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2022

Urbana, Illinois

Adviser:

Associate Professor Oluwasanmi Koyejo

ABSTRACT

Parsimony is a general guiding principle in science and philosophy which suggests that if one has multiple theories fitting the data equally well, one should choose the "simplest" theory. In the field of machine learning and artificial intelligence, the sparsity of a model is used as a measure of parsimony. Algorithms which produce an optimal set of sparse parameters for a given model have been notoriously difficult to construct due to the non-convex and combinatorial nature of sparsity constraints. In this thesis we begin by giving an overview of popular algorithms for sparse and convex optimization. We then show how they can be combined with classical tools from the theory of approximation algorithms to compute approximate projections onto the sparsity constraints, which ultimately leads to a novel algorithm for sparse optimization.

ACKNOWLEDGMENTS

First and foremost I am extremely grateful to my advisor, Professor Oluwasanmi Koyejo for his continuous support, invaluable advice, and patience over the course of my Masters study. His deep knowledge and plentiful experience have encouraged me in my academic research, and in broader aspects of my life in general. I would also like to thank Dr. Rajiv Khanna for his support on some of the more technical aspects of this work. I would like to thank all the members in the Koyejo lab group for their willingness and openness in accepting me into their research group for the past 2 years. Additionally I would like to thank my friends Damian, Addiel, and Youseff for reminding to me to enjoy life outside of academic research. Finally, I would like to express my gratitude to my parents, Malini and Sanjib, and my sister, Riona. Without their tremendous understanding and encouragement in the past few years, it would be impossible for me to complete my study.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 THE IDEAL OF SPARSITY

Parsimony is a general guiding principle in science and philosophy which suggests that if one has multiple theories fitting the data equally well, one should choose the "simplest" theory. What metric to use for the simplicity of a theory can be problem specific, but in general theories which require fewer assumptions are considered more parsimonious. In the field of machine learning and artificial intelligence, the sparsity of a model is used as a measure of parsimony. Sparse models, those which have a bounded number of non-zero parameters, have many benefits such as the prevention of overfitting, as well as maintaining the interpretability of the model. In fact, in many modern statistical estimation problems the number of parameters of the model is far greater than the number of observations, so the number of non-zero parameters of the model must be bounded to ensure consistent statistical recovery [1].

1.2 DIFFERENT APPROACHES TO SPARSITY

The difficulty of producing a sparse model stems from the combinatorial nature of sparsity constraints, as considering all subsets of a given size k would take exponential time. An intuitive approach to tackle the general intractibility of sparsity is to relax the non-convex sparsity constraint to a convex constraint which roughly captures the desire for sparse solutions. One way to go achieve this is to relax the sparsity constraint from the l_0 norm to the l_1 norm, which is convex. After this convex relaxation, assuming a convex objective function, the overall optimization problem is convex, so standard gradient descent approaches can be used to solve the relaxed problem optimally. A natural question to ask would be under what conditions does the optimal solution of the relaxed problem coincide with the optimal solution of the original problem. A sufficient condition for accurate recovery under the convex relaxation is known as the *Restricted Isometry Property* introduced by Cándes and Tao [2]. Another approach is to add a regularization term to the objective function which promotes sparsity. As an example, one common regularization term is the addition of the l_1 norm of the parameter to the objective function. Note the regularization term must be convex to ensure the overall objective remains convex. There are also many other norm choices for the convex relaxation, such as the k-support norm, which is the tightest convex relaxation of sparsity combined with a l_2 penalty [3].

1.3 TACKLING SPARSITY DIRECTLY

In recent times, there have been new developments in the field of non-convex optimization which allow algorithms to tackle the sparsity constraints directly, as opposed to using convex relaxations as a proxy for sparsity. Although considering all subsets of a given size k is algorithmically infeasible, there are situations where all subsets of a given size do not need to considered, and this can lead to efficient algorithms such as Iterative Hard Thresholding [1, 4] for sparse recovery of the model parameters. Iterative Hard Thresholding will be described in further detail in Chapter 2. The results of this thesis will follow in line with the aforementioned algorithm, as we work directly with the sparsity constraints as opposed to a convex relaxation. The remainder of this thesis is organized as follows: This chapter will conclude with a list of technical definitions required for the results presented in this work. Chapter 2 will describe the Iterative Hard Thresholding Algorithm (IHT) in more detail, as well as give perspective on viewing IHT as a sparsification of Projected Gradient Descent (PGD). Chapter 3 will describe the Mirror Descent algorithm as a generalization of PGD where the Euclidean norm is replaced with a Bregman divergence. Chapter 4 will give an overview on previous results regarding the sparsification of Mirror Descent. Finally, Chapter 5 will cover new results and algorithms for the sparsification of Mirror Descent.

1.4 TECHNICAL DEFINITIONS

Throughout this thesis, the main optimization problem of concern is of the following form

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad x \in C \cap S_k \end{aligned} \tag{1.1}$$

Where C is a convex subset of R^d and S_k is the set of sparse vectors in R^d, i.e $S_k = \{\|x\|_0 \leq k \, : \, x \in R^d\}$. Here we define $\|x\|_0$ for some n dimensional vector x as $\sum_{i=1}^{n} \mathbf{1}[x_i \neq 0]$, the number of non-zero values in the vector x. This is also referred to as the l_0 norm, although it is technically not a norm as it violates the triangle inequality. Furthermore we assume f is a continuously differentiable function which satisfies the restricted strong convexity and restricted strong smoothness properties, which will be explained in further detail below. The addition of the sparsity constraint turns the overall problem into a non-convex optimization problem, as the set S is a non-convex set.

Definition 1.1 (Convexity)**.** *A function $f : D \subseteq R^d \to R$ is convex if D is a convex set and for all $x, y \in D$ and $\alpha \in [0, 1]$ we have*

$$f\left(\alpha x + (1-\alpha)y\right) \leq \alpha f(x) + (1-\alpha)f(y) \tag{1.2}$$

Geometrically this condition ensures the line between 2 points of the function always lies above the function itself. Convex functions have been well studied for their theoretical properties with respect to optimization problems. Perhaps the most useful fact about convex functions in terms of tractability of optimization problems is that the convex condition ensures that any local minimum of the function is also a global minimum. An equivalent characterization of a convex function can be written as

$$f(y) \geq f(x) + \nabla f(x)^T (y-x) \;\; \forall x, y \in D \tag{1.3}$$

In this form the convexity condition states that a convex function can always be lower bounded by an affine function. Rearranging the above inequality yields $f(y) - f(x) \geq \nabla f(x)^T (y-x)$, which implies the function f has at least a linear rate of growth. We present the aforementioned characterization of convexity to show it can generalize to stronger notions of convexity.

Definition 1.2 (Strong Convexity [5])**.** *A function $f : D \subseteq R^d \to R$ is strongly convex with parameter $\mu > 0$ if D is a convex set and for all $x, y \in D$ we have*

$$f(y) \geq f(x) + \nabla f(x)^T (y-x) + \frac{\mu}{2}\|x-y\|^2 \;\; \forall x, y \in D \tag{1.4}$$

Clearly strong convexity implies convexity as the condition imposes a stronger lower bound. Geometrically, strong convexity ensures the function f has at least a quadratic growth rate with respect to some norm. The choice of norm is not fixed in the definition, and as we will see later, the choice of norm is important in developing algorithms for convex optimization. The degree of freedom with respect to the norm also allows us to characterize a function as convex with respect to a given norm.

Definition 1.3 (Restricted Strong Convexity (RSC) [5])**.** *A function $f : D \subseteq R^d \to R$ satisfies the RSC property with parameter $\mu > 0$ and if for all $x, y \in D$ we have*

$$f(y) - f(x) \geq \nabla f(x)^T (y-x) + \frac{\mu}{2}\|x-y\|^2 \;\; \forall x, y \in D \tag{1.5}$$

Note that the only difference between the restricted strong convexity and strong convexity conditions is that in the definition of restricted strong convexity we have dropped the requirement that the domain D of function f be a convex set. Restricted strong convexity ensures that even if the function f is not convex, we still have a quadratic lower bound on

the growth on the function on it's domain. Intuitively speaking, in the context of sparse optimization we would like the domain D to relate to the sparsity constraints so that we have a grasp on the rate of change of the function f on sparse inputs. For example in [6] they consider $D = \{x, y \,:\, \|x\|_0 \leq k\,, \|y\|_0 \leq k\,, \|x - y\|_0 \leq k\}$, i.e the set of all k-sparse vectors that differ in at most k entries. The definition of restricted strong convexity with the aforementioned k-sparse domain is the definition of restricted strong convexity we will consider for the remainder of this thesis.

Definition 1.4 (Restricted Strong Smoothness (RSS) [5])**.** *A function $f : D \subseteq R^d \to R$ satisfies the RSS property with parameter $\omega > 0$ and if for all $x, y \in D$ we have*

$$f(y) - f(x) \leq \nabla f(x)^T (y - x) + \frac{\omega}{2}\|x - y\|^2 \;\; \forall x, y \in D \tag{1.6}$$

As opposed to the lower bound on the growth of the function f imposed by the RSC condition, the RSS condition imposes a quadratic upper bound on the growth of the function f. It is also known that this condition is equivalent to a Lipschitz condition on the gradient.

Definition 1.5 (Submodularity [7])**.** *Consider an arbitrary set V. A set function $f : 2^V \to R$ is submodular if for every $A \subseteq B \subseteq V$ and $e \in V \setminus B$ it holds that*

$$f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B) \tag{1.7}$$

There are several equivalent characterizations of submodularity, but in this form, the definition of submodularity captures the fact that submodular set functions have a diminishing returns property. Given a set X and element $e \in V \setminus X$, the term $f(X \cup \{e\}) - f(X)$ can be interpreted as the marginal contribution of the element e to the value of the set function at $X \cup \{e\}$. The above condition states that if we have 2 sets $A \subseteq B$ and some element $e \in V \setminus B$, the marginal contribution of e is greater when e is added to a smaller set. In other words, the larger a set is, the smaller the marginal contribution of an additional element, hence the diminishing returns property.

Submodular set functions have been well studied in the field of theoretical computer science for their use in greedy approximation algorithms. A seminal result of Nemhauser et.al [7] showed that greedy maximization of a monotone submodular set function returns a set within a constant factor of $(1 - \frac{1}{e})$ of the optimal set of the same size. This result is the basis of many approximation algorithms related to set covering, and in the field of machine learning then concept of submodularity has been proven to be a useful concept for many tasks including sparse prediction [8] and model interpretation [9].

CHAPTER 2: IHT AS SPARSE PROJECTED GRADIENT DESCENT

2.1 ITERATIVE HARD THRESHOLDING

Consider an optimization problem of the following form

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad x \in S_k \end{aligned} \tag{2.1}$$

Where the function $f : R^d \to R$ satisfies the RSC condition on domain $D = \{x, y : \|x\|_0 \leq k\,, \|y\|_0 \leq k\,, \|x - y\|_0 \leq k\}$. Note that this optimization is similar to the main optimization problem 1.1 except for the fact we have eliminated the convex constraint on the domain of feasible solutions, so we should hope to understand how to solve an optimization problem in this form before we tackle the additional convexity constraint. As the sparsity constraint is still present, the overall problem is still non-convex, so we cannot directly apply algorithms like gradient descent and expect good convergence guarantees. Furthermore, plainly applying gradient descent would not guarantee a sparse solution even if it did converge, so more intricate algorithmic machinery is required. A simple way to force gradient descent to produce sparse solutions is to project the current iterate onto a sparse set after the gradient step. Indeed, this is exactly the intuition for the Iterative Hard Thresholding algorithm which is presented below [1].

Algorithm 2.1: Iterative Hard Thresholding

Input : function f with gradient oracle, sparsity level k, step size η

$\theta_1 = 0$

$t = 1$

repeat

  $\theta_{t+1} = P_k(\theta_t - \eta \nabla f(\theta_t))$

  $t = t + 1$

until *stop criteria*

Output: θ_t

Where $P_k(.)$ is the Euclidean projection operator onto S_k, i.e $P_k(x) = \text{argmin}_{y \in S_k} \|x - y\|_2^2$. Since we are using the Euclidean norm to measure the distance between a point and it's projection, the corresponding projection is computationally easy to compute. Given some vector $x \in R^d$ we would like to compute the projection of, consider a permutation σ on the indices of x such that $x_{\sigma(1)} > x_{\sigma(2)} > ... > x_{\sigma(d)}$. Recall that $\|x\|_2^2 = \sum_{i=1}^d x_i^2$. In this form

it is easy to see $\|x-y\|_2^2$ subject to $\{y \in S_k\}$ is minimized when $y_{\sigma(i)} = x_{\sigma(i)}$ for $i \in [k]$, and $y_{\sigma(i)} = 0$ otherwise. In other words, the projection of x under the Euclidean norm can be computed by simply picking the top k elements in terms of magnitude, and setting the rest of the elements to 0, which is also known as the hard thresholding operator [10].

Lemma 2.1 (IHT Convergence, [1]). *Let f have RSC and RSS parameters μ and ω respectively and let the IHT algorithm be invoked with the function f, $\hat{k} = 32(\frac{\mu}{\omega})^2 k$, and let $\eta = \frac{2}{3\mu}$. Let $^* = arg\ min_{x \in S_k} f(x)$. Then on the τ-th iteration of the IHT algorithm for $\tau = O(\frac{\mu}{\omega} \log \frac{f(x_0)}{\epsilon})$ satisfies*

$$f(x^\tau) - f(x^*) \leq \epsilon \tag{2.2}$$

The lemma above states the main convergence result of the Iterative Hard Thresholding algorithm in the context of M-estimation [1]. The surprising fact to take note of is that although the overall optimization problem is non-convex, we are still able to converge to a global optimum without the use of any convex relaxation, but rather by using the non-convex constraint directly. This is possible due to the fact that sparsity constraints are more structured then general non-convex constraints, and the fact that the projection under the Euclidean norm can be computed exactly. A more detailed discussion of the projection operator is presented below.

2.2 PROJECTION OPERATOR

In the vanilla Iterative Hard Thresholding algorithm presented above, we project onto the space S_k, but in general we could encode more structure into the sparsity constraint. For example, we could consider the notion of *group sparsity*, where we have a set of supports $G = \{G_1, G_2, ..., G_m\}$ where $G_i \subseteq [d]$ and a set of sparsity levels $\{k_1, k_2, ..., k_m\}$ for each support. Let x_{G_i} denote the elements of x under the support G_i. The sparsity constraints can be written as $\|x_{G_i}\|_0 \leq k_i \ \forall i \in [m]$. For simplicity consider the case when $G_i \cap G_j = \emptyset$ for all $i, j \in [m]$. This is known as the non-overlapping case, as the supports are disjoint and thus non-overlapping. The projection operator for the non-overlapping group sparsity constraint differs from the projection operator in the vanilla sparsity constraint case, but can still be efficiently computed by projecting thresholding the top k_i values for each group support x_{G_i}. One way to view the Iterative Hard Thresholding algorithm is just as a special case of projected gradient descent, and like all projected gradient descent algorithms the efficiency of the overall algorithm is heavily dependent on the efficiency of the projection operator. Therefore many new results which modify the Iterative Hard Thresholding algorithm for

different constraint sets focus on the the projection operator as the main mathematical object of study. In the case where the groups are allowed to overlap, an exact projection cannot be computed and instead we must rely on approximate projections to the sparsity constraint set. It is interesting to note that the approximate projection operator in the overlapping group case relies on reducing the projection to a submodular maximization problem, which is yet another example of the effectiveness of submodularity as an algorithmic tool in machine learning [11]. This result is also relevant since it uses an approximate projection as opposed to an exact projection at each iteration, which will become a more central theme in latter sections.

2.3 GRADIENTS AND DUAL NORMS

While not obvious at first glance, there is a technical point to be made about the projection step, specifically regarding why we are able to subtract the gradient from the current iterate directly like so $P_{s_k}(\theta_t - \eta\nabla f(\theta_t))$, as the input to the projection operator. To explore this technicality, we must first understand the notion of a dual norm.

Definition 2.1 (Dual Norm)**.** *Given some norm $\|.\|$ defined on R^d, we define the dual norm $\|.\|_*$ as a function from $R_d \to R$ with values*

$$\|z\|_* = \max_x (x^T z) \ : \ \|x\| \leq 1 \tag{2.3}$$

For example, the dual norm to the 1-norm can be computed as $\|z\|_* = \max_{\|x\|_1 \leq 1}(x^T z)$. We have that $z^T x \leq \sum |z_i x_i| \leq \sum |z_i||x_i| \leq max_i|z_i| \sum |x_i| \leq max_i|z_i|$ since $\|x\|_1 \leq 1$. So we have that $\max_x z^T x \leq \|z\|_\infty$, and we can achieve this bound by choosing $x = sign(z_i)e_i$ where z_i is the component in z with maximum norm. This yields $\|z\|_* = \max_{\|x\|_1 \leq 1}(x^T z) = \max_i z_i = \|z\|_\infty$. Therefore, the dual of the 1-norm is the ∞-norm. In general the dual of a p-norm can be computed as a q-norm such that $(\|x\|_p)_* = \|x\|_q$ where $\frac{1}{q} + \frac{1}{p} = 1$ and we adopt the convention that $\frac{1}{\infty} = 0$. Using the aforementioned fact, we see that the dual of the 2-norm is the 2-norm itself, so the 2-norm is self dual. Related to this fact about the self-duality of the 2-norm, by the Riesz representation theorem, given some Hilbert space $\mathcal{H}$, the dual space $\mathcal{H}^*$ is isometric to $\mathcal{H}$ itself. Now consider the general case when we are optimizing in some Banach space $\mathcal{B}$ ($\mathcal{B} = l_1$ as an example). In this setting, subtracting the gradient term directly from the iterate as is done in the gradient descent algorithm, projected gradient descent and IHT algorithms, is not a formally defined mathematical operation. This is because the gradient is a linear functional, so technically the elements of $\nabla f(x)$ do not lie in the primal space $\mathcal{B}$, but in the dual space $\mathcal{B}^*$, so the operation $\theta_t - \eta\nabla f(\theta_t)$ is not defined

since θ_t exists in the primal space $\mathcal{B}$ and $\nabla f(\theta_t)$ exists in the dual space $\mathcal{B}^*$. The reason we are able to directly subtract the gradient from the iterate in the case of an Euclidean space is because the dual space is isometric, but for more general optimization problems this may not hold. Further details on this discussion are given in [12]. To address this problem in the context of convex optimization, we will introduce the well known algorithm of Mirror Descent.

CHAPTER 3: MIRROR DESCENT

3.1 PROBLEM INTRODUCTION

In the previous section we saw an algorithm to tackle the optimization problem which only includes the sparsity constraint. We will now address optimization problem which only includes the convex constraint, and along the way we will see how this addresses the aforementioned issue in the previous chapter with primal and dual spaces. Formally, consider the following optimization problem

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad x \in C \end{aligned} \tag{3.1}$$

where C is some convex set, f is a continuously differentiable convex function. This is a classical optimization problem with applications in both theoretical and applied domains, and as a result has been extremely well studied. As we have seen, the Projected Gradient Descent algorithm may not be optimal in all situations to solve this problem. Before we introduce the celebrated Mirror Descent algorithm [13], some preliminaries are required.

Definition 3.1 (Convex Conjugate)**.** *Given a function $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$, the convex conjugate $f^* : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is defined as*

$$f^*(y) = \sup_{x \in R^n} (y^T x - f(x)) \tag{3.2}$$

Given an input y, $f^*(y)$ is the maximum value by which the linear function of y exceeds $f(x)$. Alternatively, one can think of $f^*(y)$ as the value that $y^T x$ must be shifted until it is a support of f. Geometrically, the convex conjugate is a representation of the function f as a set of tangent hyperplanes, and the parameters of these hyperplanes are encoded in the conjugate function f^*.

Example 3.1. *Let $f(x) = cx$ for some $c \in \mathbb{R}$. Then the conjugate function can be computed as follows*

$$f^*(y) = \sup_{x \in \mathbb{R}}(yx - cx) = \sup_{x \in \mathbb{R}}((y - c)x) \tag{3.3}$$

$$\implies f^*(y) = \begin{cases} 0 & \text{if } y = c \\ \infty & \text{otherwise} \end{cases} \tag{3.4}$$

Therefore, the convex cojugate of $f(x) = cx$ is $f^*(y) = \delta_c$, the indicator function for c. More relevant to our setting, given a convex set C, let δ_C be the indicator function for the convex set C defined as follows

$$\delta_C(x) = \begin{cases} 0 & x \in C \\ \infty & x \notin C \end{cases} \tag{3.5}$$

Example 3.2. *Given a convex set C, let $f(x) = \delta_C(x)$, the indicator function for the convex set. Then the conjugate function can be computed as follows*

$$f^*(y) = \sup_{x \in \mathbb{R}^n} (y^T x - \delta_C(x)) = \sup_{x \in C}(y^T x) \tag{3.6}$$

If $x \notin C$, then the quantity $y^T x - \delta_C(x)$ is arbitrarily negative, so we only have to consider values of $x \in C$, in which case $\delta_C(x) = 0$. The conjugate function in this case is known as the support function of the set C, since it defines a set of supporting hyperplanes to the set C.

On may notice some similarities between the definition of the convex conjugate and the definition of a dual norm, and indeed they are related in the following way

Example 3.3. *Let $f(x) = \|x\|$ for some norm $\|.\|$. Then the conjugate function can be computed as follows*

$$f^*(y) = \sup_{x \in \mathbb{R}}(y^T x - \|x\|) \tag{3.7}$$

$$\implies f^*(y) = \begin{cases} 0 & \|y\|_* \leq 1 \\ \infty & \|y\|_* > 1 \end{cases} \tag{3.8}$$

In other words, the convex conjugate of a norm function is an indicator function for the unit ball of the dual norm. Another important concept to understand as a precursor to understanding the Mirror Descent algorithm is the Bregman divergence.

Definition 3.2 (Bregman Divergence [13])**.** *Le f be a continuously differentiable convex function. The Bregman Divergence between 2 points $x, y \in Dom(f)$ is defined as*

$$B_f(x, y) = f(x) - f(y) - \langle \nabla f(y), x - y \rangle \tag{3.9}$$

For an intuition of this quantity, recall the first order Taylor approximation of a function at a point can be written as $f(x) \approx f(y) + \langle \nabla f(y), x - y \rangle$. With this in mind the Bregman

divergence can be seen as the difference between the true value of the function f at x and the value of the linear approximation of f centered at the point y. Let us now see a few examples of functions and the corresponding induced Bregman Divergence.

Example 3.4. *Define $f : R^d \to R$ as $f(x) = \|x\|_2^2$. Then the induced Bregman divergence is*

$$B_f(x, y) = f(x) - f(y) - \langle \nabla f(y), x - y \rangle \tag{3.10}$$
$$= \|x\|_2^2 - \|y\|_2^2 - 2\langle y, x - y \rangle \tag{3.11}$$
$$= \|x\|_2^2 + \|y\|_2^2 - 2\langle y, x \rangle \tag{3.12}$$
$$= \|x - y\|_2^2 \tag{3.13}$$

Therefore, the l_2 norm function induces the Euclidean distance as it's corresponding Bregman divergence.

Example 3.5. *Define $f : R^d_{\geq 0} \to R$ as $f(x) = \sum_{i=1}^n x_i \ln x_i$, otherwise known as the negative entropy function. Then the induced Bregman divergence is*

$$B_f(x, y) = f(x) - f(y) - \langle \nabla f(y), x - y \rangle \tag{3.14}$$
$$= \sum_{i=1}^n x_i \ln x_i - \sum_{i=1}^n y_i \ln y_i - \sum_{i=1}^n (\ln y_i + 1)(x_i - y_i) \tag{3.15}$$
$$= \sum_{i=1}^n x_i \ln x_i - \sum_{i=1}^n x_i \ln y_i - \sum_{i=1}^n (x_i) + \sum_{i=1}^n (y_i) \tag{3.16}$$
$$= \sum_{i=1}^n x_i \ln \frac{x_i}{y_i} - \sum_{i=1}^n (x_i) + \sum_{i=1}^n (y_i) \tag{3.17}$$
$$= D_{KL}(x \| y) \tag{3.18}$$

Therefore, the negative entropy function induces the generalized KL-divergence as it's corresponding Bregman Divergence. In the special case when $\sum x_i = \sum y_i = 1$, this reduces to the ordinary KL divergence.

Recall we began this discussion of the Mirror Descent algorithm as a remedy for the potential shortcomings of gradient descent, so in some sense we should expect Mirror Descent to generalize gradient descent type algorithms. As discussed in the previous section, the iterative update for Projected Gradient Descent can be written as

$$x_{k+1} = P_C(x_k - \eta \nabla f(x_k)) \tag{3.19}$$

Where $P_C(.)$ is the Euclidean projection operator. By the definition of Euclidean projection we have

$$x_{k+1} = \arg\min_{x \in C} \|(x_k - \eta \nabla f(x_k)) - x\|_2^2 \tag{3.20}$$

$$= \arg\min_{x \in C} \|(x - x_k) + \eta \nabla f(x_k)\|_2^2 \tag{3.21}$$

Since the l_2 norm is induced by an inner product, we can expand as follows

$$\|(x - x_k) + \nabla \eta f(x_k)\|_2^2 = \|x - x_k\|_2^2 + 2\eta\langle x, \nabla f(x_k)\rangle + 2\eta\langle -x_k, \nabla f(x_k)\rangle + \|\eta \nabla f(x_k)\|_2^2 \tag{3.22}$$

Note that the last 2 terms of the expansion do not depend on x, so therefore they can be excluded from the optimization. Rearranging the remaining terms yields

$$x_{k+1} = \arg\min_{x \in C} \left\{ \langle x, \nabla f(x_k)\rangle + \frac{\|x - x_k\|_2^2}{2\eta} \right\} \tag{3.23}$$

In this form, we can interpret the next iterate x_{k+1} as the point which strikes the best balance between following the direction of steepest descent, represented by the inner product term, and not straying too far from the current iterate, represented by the proximal term, while still remaining in the constraint set C. Since we started with the Euclidean projection operator, we see that the proximal term uses the l_2 norm as a measure of distance between points. A natural question that should arise is whether we can generalize the proximity term to other measures of distance. Using $d(.\|.)$ as a placeholder for an arbitrary distance function, we recover the *Generalised Projected Gradient Descent* algorithm.

$$x_{k+1} = \arg\min_{x \in C} \left\{ \langle x, \nabla f(x_k)\rangle + \frac{d(x, x_k)}{2\eta} \right\} \tag{3.24}$$

From an optimization perspective, we must know ask the following question: For what choices of $d(.\|.)$ is the aforementioned optimization problem feasible? As one may expect based on the definitions presented in the beginning of the chapter, we can select $d(.|\ .)$ to be a Bregman divergence and solve for a closed form iteration for x_{k+1}. Say we have a μ strongly convex function ψ, let B_ψ be the associated Bregman divergence, then the resulting iteration becomes

$$x_{k+1} = \arg\min_{x \in C} \left\{ \langle x, \nabla f(x_k)\rangle + \frac{B_\psi(x, x_k)}{2\eta} \right\} \tag{3.25}$$

For a closer look at the derivation of the closed form iteration, refer to the work of Beck and

Teboulle [13], which was one of the first works to introduce the Mirror Descent algorithm. The full Mirror Descent algorithm is presented below.

Algorithm 3.1: Mirror Descent

Input: $y_1 \in$ dom $\nabla\psi^*$

repeat

$x_k = \nabla\psi^*(y_k)$

$y_{k+1} = \nabla\psi(x_k) - \mu_k \nabla f(x_k)$

$x_{k+1} = \nabla\psi^*(y_{k+1})$

until *stop criteria*

Where ψ^* is the convex conjugate of ψ.

3.2 BENEFITS OF MIRROR DESCENT

Recall we introduced the Mirror Descent algorithm as an algorithm for solving the optimization problem presented at the beginning of this chapter, namely minimizing a convex function under a convex constraint set. We have alluded to the benefits of Mirror Descent over Projected Gradient Descent, but in this chapter we will go into further details regarding these benefits.

3.2.1 Differentiation of Primal and Dual Space

One of the main issues we presented in the previous chapter regarding Projected Gradient Descent algorithms in general, is that they do not differentiate the primal space the iterate x_k exists in and the dual space that $\nabla f(x_k)$ exists in. We can justify conflating $\mathbb{R}^n$ equipped with the l_2 norm with it's dual space as they are isometric, but as mentioned in the previous chapter, this justification is not sufficient for more general spaces. One major benefit of Mirror Descent is that it explicitly distinguishes between the primal and dual spaces, and specifies a bijection we can use to map between these spaces. Specifically this bijection is determined by our choice of function $\psi : R^n \to R$. The bijection will map the primal point x to it's dual point $\nabla f(x_k)$, and to map from the dual space back to the primal space we use the inverse mapping of $x \to \nabla f(x)$, which is given by, $y \to \nabla\psi^*(y)$, the dual of ψ. With this in mind, the steps of the Mirror Descent algorithm can be interpreted as follows: We start with some feasible point in the primal space, we then project the feasible point onto the dual gradient space, we take a step in this dual space, then finally we project back to

the primal space to recover the next iterate.

3.2.2 Improving Dimension Dependence

Say we run the Gradient Descent algorithm for T steps, and let x_i denote the iterate at the i-th step. Let $\epsilon_i = f(x_i) - f(x^*)$, the difference between the functional value at the current iterate and the functional value at the optimal at the i-th iteration. It is known that Gradient Descent converges in $O(\frac{1}{\sqrt{T}})$ time, and specifically we have that

$$\min_{i \in [T]} \epsilon_i \leq \frac{RG}{\sqrt{T}} \tag{3.26}$$

where $\|\nabla f(x)\|_2 \leq G$ for all $x \in C$ and $R = \max_{x \in C} \|x_1 - x\|_2$. G can be interpreted as an upper bound on the norm of the gradient for all feasible values of x, and R can be thought of as the diameter of the constraint set C. In the above form, the dimension dependence is not easy to see, but consider the following example when C is the probability simplex, so we have $C = \{x \in R^n : \|x\|_1 = 1\}$. In this setting, $R \leq \sqrt{2}$, and if each coordinate of the gradient ∇f_i is bounded by M, then we have that $G \leq M\sqrt{n}$, so G depends on the dimension, which does not scale well for high dimensional problems.

The convergence bound for Mirror Descent is similar, but with a few key differences. First, the upper bound on the gradient is measured in terms of the dual norm, so $\|\nabla f(x)\|_* \leq G$, and second the diameter of the constraint set is measured using a Bregman divergence. Intuitively, one should should pick a Bregman divergence that measures the diameter of the convex constraint set in a nice way. and in this way one can fine tune the Mirror Descent algorithm to the geometry of the specific problem at hand. In our example given C is the simplex, say we choose ψ to be the negative entropy function, so we induce the KL-Divergence as the corresponding Bregman divergence. Note that ψ is 1-strongly convex with respect to the l_1 norm. Recall the dual of the l_1 norm is the l_∞ norm. and assuming each coordinate of the gradient ∇f_i is bounded by M as above, we can bound $\|\nabla f(x)\|_* \leq M = G$. By measuring the gradient in the dual norm, we have removed the dimension dependence in G. If we set $x_1 = \frac{1}{n}\mathbf{1}$, the uniform distribution vector, then we can bound $B_\psi(x_1, x^*) \leq \log n$ since B_ψ is the KL-Divergence. This yields a final value of $M \log n$ in the numerator of the convergence bound for Mirror Descent, which compared to the bound $M\sqrt{2n}$ for gradient descent gives an improvement of order $O\left(\sqrt{\frac{n}{\log n}}\right)$. Therefore, by choosing a Bregman divergence which reflects the geometry of the constraint set, Mirror Descent can converge to an optimal solution faster than Gradient Descent.

CHAPTER 4: PAST RESULTS ON MIRROR DESCENT SPARSIFICATION

Now that sufficient background on Mirror Descent and sparse optimization have been developed, let us consider ways to combine these 2 concepts together. In this section we will cover some previous results on the use of Mirror Descent for sparse optimization problems. Recall the overall optimization problem 1.1 we are interested in solving

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad x \in C \cap S_k \end{aligned} \tag{4.1}$$

Furthermore, recall some of the methods mentioned in Chapter 1, such as convex relaxation and regularization, which are used to solve these sparse optimization problems indirectly, without explicitly working with the sparsity constraints. One may consider combining these methods when with Mirror Descent to induce sparse solutions, and indeed this is what past results on Mirror Descent for sparse optimization do.

4.1 SMIDAS ALGORITHM

A noteable result in the this problem area is a result of Shalev-Shwartz and Tewari [14], in which they introduce the Stochastic Mirror Descent Made Sparse (SMIDAS) algorithm. The SMIDAS algorithm is a modified version of Mirror Descent which solves the l_1 regularized optimization problems. Specifically, the optimization problems they consider are of the form

$$\min_{w \in R^d} \frac{1}{m} \sum_{i=1}^{m} L(\langle w, x_i \rangle, y_i) + \lambda \|w\|_1 \tag{4.2}$$

This can be related to the original optimization problem we are interested in by considering $C = R^n$ and setting $f(x) = \frac{1}{m} \sum_{i=1}^{m} L(\langle w, x_i \rangle, y_i)$. Statistical problems where the objective function is an average of samples like above are referred to as M-estimation problems. Without the use of the regularization term in the objective function, we know we can solve the optimization problem by plainly applying the Mirror Descent algorithm. An initial way one might consider modifying the Mirror Descent algorithm for the regularization term is to subtract the gradient of $\lambda \|w\|_1$ from the dual vector before projecting back into the primal space. Technically since the l_1 norm is not differentiable, we must use a subgradient of $\|w\|_1$, for example a vector whose i-th element is equal to $sign(w_i)$, where we consider $sign(0) = 0$. Let us denote such a vector as $sign(w)$. Recall the original gradient update of Mirror Descent is of the form $y_{k+1} = \nabla \psi(x_k) - \mu_k \nabla f(x_k)$. Factoring in the regularization

term, the new update would become $y_{k+1} = \nabla\psi(x_k) - (\mu_k(\nabla f(x_k) + \lambda sign(w)))$. Unfortunately as noted by Langford et.al [15], this will actually lead to a dense vector in the dual, which will consequently lead to a dense primal vector, which is clearly undesirable as we want Mirror Descent to produce sparse solutions. Alternatively, this paper proposes a breaking up the gradient step into 3 separate steps. The first step is the standard gradient step $y_{\frac{1}{2}} = \nabla\psi(x_k) - \mu_k\nabla f(x_k)$. The next step is the gradient step for the regularization term computed on the gradient after the first step, $y_1 = y_{\frac{1}{2}} - \mu_k\lambda sign(y_{\frac{1}{2}})$. Finally, there is a truncation step where if $sign(y_{\frac{1}{2},i}) \neq sign(y_{1,i})$, then we set $y_{1,i}$ to 0 where $_{1,i}$ is the i-th component of y_1. Intuitively, the first gradient step is taken to minimize the objective function, and the next 2 steps are to minimize the regularization term, with the truncation step at the end to promote sparse solutions. The algorithm for this procedure is presented below.

Algorithm 4.1: SMIDAS

Input: $\mu \geq 0$

Let $x_0 = 0$

repeat

Sample i uniformly at random from $[m]$

$y_{k+\frac{1}{2}} = \nabla\psi(x_k) - \mu_k\nabla f(x_i)$

$\forall j : y_{k+1,j} = sign(y_{k+\frac{1}{2},j}) \times \max\{0, |y_{k+\frac{1}{2},j}| - \mu\lambda\}$

$x_{k+1} = \nabla\psi^*(y_{k+1})$

until *stop criteria*

Note that the a sample is picked uniformly at random to compute the gradient as opposed to using the entire dataset each iteration, so this is really a modification of the Stochastic Mirror Descent algorithm. While this algorithm does not consider the sparsity constraints directly, modifying the standard Mirror Descent algorithm for an l_1 regularized optimization problem is still an important result for the use of Mirror Descent in sparse optimization. It is interesting to note that even though the sparsity constraint is imposed on the primal vector, the SMIDAS algorithm is able to achieve sparse solutions by making the dual vector more sparse. Inspired by the SMIDAS algorithm of, a possible algorithm one may come up with is to threshold the dual vector, only keeping the top k values of the dual vector and setting the rest to 0, before projecting back into the primal space. The main issue with that idea is that we sparsify the dual vector before projecting back into the primal space, and even in the simple case of Euclidean projections, a sparse vector y_s does not imply that the projection

$x = P_C(y_s)$ will be sparse. In general sparsity in the dual space does not imply sparsity in the primal space, so the only way to guarantee sparsity in the primal space is to work with the sparsity constraints in the primal space directly. How exactly we will accomplish this given the intractable combinatorial nature of sparsity constraints will be elaborated in the next section.

CHAPTER 5: NEW RESULTS ON MIRROR DESCENT SPARSIFICATION

Now that the relevant preliminaries have been covered, we can now begin the discussion on new resulting regarding the use of Mirror Descent for sparse optimization. Recall the optimization problem (1.1) of interest in this thesis.

$$\begin{aligned} &\text{minimize} && f(x) \\ &\text{subject to} && x \in C \cap S_k \end{aligned} \tag{5.1}$$

We will first give an overview of the new algorithm proposed in this thesis for the modification of Mirror Descent for sparse optimization, and then go into the technical details and proofs regarding the algorithm. The ending of the last chapter alluded to the fact that if we want to work with the sparsity constraint directly, we will need to do so in the primal space as opposed to the dual. An initial idea one may have based upon this is to threshold the primal vector at the end of each iteration, similar in style to the Iterative Hard Thresholding algorithm presented in Chapter 2. The issue with this approach is that we cannot simply pick the top k values of the primal vector as the projection onto the sparse set S_k. The reason this approach worked in the case of Iterative Hard Thresholding was because choosing the top k values is the optimal projection onto S_k using the l_2 norm, but since Mirror Descent generalizes beyond Euclidean metrics, we cannot assume that thresholding the top k values is optimal. As computing the exact projection in the general case is NP-hard, we will instead use an approximate projection inspired by approximation algorithms for set covering in classical computer science. We will show that we can bound the quality of the approximation, given certain conditions on the constraint set C, and that an approximate projection is sufficient for convergence to the optimal solution.

5.1 INITIAL PROOF OF ITERATION VIA DESCENT LEMMA

Recall the proximal form of the Mirror Descent iteration.

$$x_{k+1} = \arg\min_{x \in C} \left\{ \langle x, \nabla f(x_k) \rangle + \frac{B_\psi(x, x_k)}{2\eta} \right\} \tag{5.2}$$

If we want to consider an exact expression for the optimal projection of the next iterate onto the sparse set S_k, we can write this as

$$x_{k+1} = \arg\min_{x \in C \cap S_k} \left\{ \langle x, \nabla f(x_k) \rangle + \frac{B_\psi(x, x_k)}{2\eta} \right\} \tag{5.3}$$

It is known that optimization problems of the above form are NP hard, even in the case of "simple" loss functions like the squared loss. Before we discuss how to approximate a solution to this subproblem, we must first show that this is indeed the correct subproblem to solve. Assume we could solve this subproblem exactly, would the corresponding iterates converge to the optimal solution? The next series of proofs, based on the Descent Lemma of [16] proves that this is indeed the case. Note that in it's original form, the Descent Lemma assumes the functions has a L Lipschitz continous gradient, but this is equivalent to the Restricted Strong Smoothness property with parameter L.

Lemma 5.1 (Descent Lemma)**.** *Assume f satisfies the Restricted Strong Smoothness property with parameter L_f , with respect to some norm $\|.\|$ is a continuously differentiable function. Then for all $L \geq L_f$, and all $x, y \in R^n$ the standard descent lemma states*

$$f(x) \leq h_L(x, y) = f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2}\|x - y\|^2 \tag{5.4}$$

Where $h_L(x, y)$ is a shorthand for the smoothness inequality parameterized by L. Now assume ψ is also L strongly convex function with respect to the same norm as above for some $L \geq L_f$, i.e

$$\psi(x) \geq \psi(y) + \langle \nabla \psi(y), x - y \rangle + \frac{L}{2}\|x - y\|^2 \tag{5.5}$$

Let B_ψ be the Bregman Divergence associated with the function ψ. We can now generalize the aforementioned Descent Lemma from the norm operator to Bregman Divergences.

Lemma 5.2 (Bregman Descent Lemma)**.** *Assume f satisfies the Restricted Strong Smoothness property with parameter L_f , with respect to some norm $\|.\|$ is a continuously differentiable function. Then for all $L \geq L_f$, given a function ψ that is L-strongly convex, for all $x, y \in R^n$ we have*

$$f(x) \leq h_{LB}(x, y) = f(y) + \langle \nabla f(y), x - y \rangle + B_\psi(x, y) \tag{5.6}$$

Where $h_{LB}(x, y)$ is a shorthand for the smoothness inequality with a Bregman divergence B_ψ induced by a function ψ, which is L-strongly convex

Proof (5.1): By the definition of Bregman Divergence we have

$$B_\psi(x,y) = \psi(x) - \psi(y) - \langle \nabla\psi(y), x-y\rangle \tag{5.7}$$

$$\geq \psi(y) + \langle \nabla\psi(y), x-y\rangle + \frac{L}{2}\|x-y\|^2 - \psi(y) - \langle \nabla\psi(y), x-y\rangle \tag{5.8}$$

$$= \frac{L}{2}\|x-y\|^2 \tag{5.9}$$

Where the first inequality uses the fact that ψ is L-strongly convex.

We will now prove that this iterative scheme converges to the optimal solution.

Theorem 5.1 (Fixed Point Iteration Convergence)**.** *Let x^* be the optimal solution to optimization problem 1.1. Then x^* is a fixed point of the iterative scheme, so we have that*

$$x^* = \arg\min_{x\in C\cap S}\left\{\langle x, \nabla f(x^*)\rangle + \frac{1}{\alpha_k}B_\psi(x,x^*)\right\} \tag{5.10}$$

Proof (5.2): Recall the proposed iterative scheme from above

$$x_{k+1} = \arg\min_{x\in C\cap S}\left\{\langle x, \nabla f(x_k)\rangle + \frac{1}{\alpha_k}B_\psi(x,x_k)\right\} \tag{5.11}$$

Note that the point $x_{k+1} \in \arg\min_{z\in C\cap S}\{h_{LB}(z,x_k)\}$ is equivalent to the point x_{k+1} in the above iteration, since the terms which appear in $h_{LB}(z,x)$, but do not appear in the above iteration only depend on x_k, which is constant at each iteration.

We know that $h_{LB}(x_{k+1},x_k) \leq h_{LB}(x_k,x_k) = f(x_k)$ by the optimality of x_{k+1}. Let us now define a Bregman divergence B_π, induced by a function π, which is L_f-strongly convex. By the Bregman Descent Lemma, we have

$$f(x_k) - f(x_{k+1}) \geq f(x_k) - h_{L_f,b}(x_{k+1},x_k) \tag{5.12}$$

By definition, we have $h_{L_f,b}(x,y) = h_{LB}(x,y) - (B_\psi(x,y) - B_\pi(x,y))$. Recall that we assume the function f is L_f strongly smooth, and in order for the Bregman Descent Lemma 5.2 to hold, we need ψ to be at least L_f strongly convex, so B_π can be thought of as the Bregman divergence induced by the π with the minimum curvature needed for iterative improvement. Combined with the above property of h_{LB}, this yields

$$f(x_k) - f(x_{k+1}) \geq B_\psi(x_k,x_{k+1}) - B_\pi(x_k,x_{k+1}) \geq \frac{L-L_f}{2}\|x_{k+1}-x_k\|^2 \tag{5.13}$$

This not only shows that the proposed iterative scheme makes progress at each iteration, it also proves that the optimal point $x^* \in \arg\min_{x \in C \cap S_k} f(x)$ is a fixed point of the iteration i.e

$$x^* = \arg\min_{x \in C \cap S} \left\{ \langle x, \nabla f(x^*) \rangle + \frac{1}{\alpha_k} B_\psi(x, x^*) \right\} \tag{5.14}$$

because if it were not, we could apply the above iterative inequality, but that would contradict the optimality of x^*.

The remaining issue is now how to solve the subproblem in each iteration, i.e

$$x_{k+1} = \arg\min_{x \in C \cap S} \left\{ \langle x, \nabla f(x_k) \rangle + \frac{1}{\alpha_k} B_\psi(x, x_k) \right\} \tag{5.15}$$

5.2 APPROXIMATING SPARSE SOLUTIONS WITH SUBMODULARITY

As suggested in the introduction for this section, the above optimization subproblem takes exponential time to solve exactly, unless $P = NP$. Therefore it is not unreasonable to shift our view from exact solutions to approximate solutions. It is not hard to see that finding the best set of k atoms to project to reduces to a set selection problem, so we should shift our attention to approximation algorithms for related problems such as set covering. In a seminal result by Nemhauser [7], it was proved that for a set selection problem with a monotone submodular cost function on the sets, a greedy algorithm will produce a set of k sets that approximates the optimal choice of k sets to a factor of $(1 + \frac{1}{e})$. In order to apply the aforementioned result on greedy selection algorithms, we need some sort of set function, i.e for some set X, a function $f : 2^{|X|} \to R$ that assigns each subset to a number corresponding to the cost of the set. A natural way to do this is to create set function for each set of atoms and assign cost based on improvement to objective. In addition, in order to directly apply the result of Nemhauser, we would also need the defined set function to be submodular. Thankfully, a result of Das and Kempe [17] proves that as long as we can bound the *submodularity ratio*, which we will define later, we can still obtain a constant factor approximation, assuming fixed convexity and smoothness parameters. Furthermore, a result of Elenberg et al. [18] shows that strong convexity implies a bounded submodularity ratio, which allows us to apply the greedy selection algorithm and obtain a constant factor approximation to the optimal. The only issue is the result of Elenberg et.al does not consider the case of sparse optimization problems with convex constraints, so in the following sections we prove that under certain conditions regarding the geometry of the constraints, we can

also bound the submodularity ratio and obtain a constant factor approximation algorithm.

Definition 5.1 (Set Function). *Let us define a set function on a set X as $f : 2^{|X|} \to \mathbb{R}$*

$$\max_{S:|S|\leq k} f(S) = \max_{\substack{\beta:\beta_{S^c}=0 \\ \beta\in C \\ |S|\leq k}} l(\beta) - l(0) \tag{5.16}$$

In the above definition, β can be thought of as the model parameters and the input to the set function f can be thought of as a support set, so given some a potential support set S, β is the set of parameter values which maximizes the loss function constrained by the support set S. In our problem setting, the loss function is equal to the function value of the subproblem, so we have $l(x) = \langle x, \nabla f(x)\rangle + \frac{1}{\alpha_k} B_\psi(x, x_k)$. Note the difference between the set function defined here and the one defined in [18] is the additional constraint that β is now constrained to some convex set C. Let β^X be the β which maximizes $f(X)$, and let B_j^X denoted the jth component. We now proceed to attempt to lower bound the submodularity ratio defined as follows

Definition 5.2 (Submodularity Ratio, Weak Submodularity). *Let $S, L \subset [d]$ be 2 disjoint sets, and $f : 2^d \to \mathbb{R}$. The submodularity ratio of L with respect to S is given by*

$$\gamma_{L,S} = \frac{\sum_{j\in S} [f(L \cup \{j\} - f(L))]}{f(L \cup S) - f(L)} \tag{5.17}$$

The submodularity ratio of a set U with respect to an integer k is given by

$$\gamma_{U,k} = \min_{\substack{L,S:L\cap S=\emptyset \\ L\subseteq U, |S|\leq k}} \gamma_{L,S} \tag{5.18}$$

As alluded to previously, we need to assume a condition on the geometry of the constraint set which will allow us to lower bound the submodularity ratio. As before for some S, let β^S be the set of parameters which maximize $f(S)$ and let β_j^S be the vector where only j-th index has a value equal to the j-th index of β^S, and all other indices have value 0. We will first present the definition, then give an intuitive explanation of the restriction this condition places on the geometry of the constraint set.

Definition 5.3 (Incremental Update Condition). *Given some set L such that $|L| < k$, there must exist some set X such that $L \subset X$, $j \in X$, $j \notin L$ and the following condition holds*

$$\frac{\|\nabla l(\beta^L)\|_2^2}{M_{L+1}} < \|\beta_j^X - \beta^L\|_2^2 \tag{5.19}$$

First note that given some set L, the value of $\frac{\|\nabla l(\beta^L)\|}{M_{L+1}}$ is constant. In the language of submodularity, the term on the right hand side can be viewed as the marginal benefit gained by adding the atom j into the support set. So given some set L, this condition places a lower bound on the marginal benefit of adding a new element into the support. The technical purpose of this condition will be elucidated in the proof of the lower bound of the submodularity ratio, but intuitively to bound the submodularity ratio, we need to ensure we make enough progress with each greedy selection, and this condition places a lower bound on the the progress we make with each selection.

Theorem 5.2. *Define $f(S)$ as above in definition 5.1, and assume $l(.)$ is a $(m_{|U|+k}, M_{|U|+k})$ (strongly concave, smooth) function on domain $\Omega_{|U|+k}$ and $\tilde{M}_{|U|+1}$ smooth on $\Omega_{|U|+1}$. Also assume the Incremental Update Condition (5.3) holds. Then we have that $\gamma_{U,k}$ is lower bounded by*

$$\gamma_{U,k} \geq \frac{m_{|U|+k}}{\tilde{M}_{|U|+1}} \geq \frac{m_{|U|+k}}{M_{|U|+k}} \tag{5.20}$$

Proof (5.3):
Let $\gamma > 0$. We call a function γ-weakly submodular at a set U and an integer k if $\gamma_{U,k} \geq \gamma$ We first upper bound the denominator of the $\gamma_{L,S}$. Let $\overline{k} = |L| + k$. Applying the definition of strong concavity we get

$$\frac{m_{\overline{k}}}{2}\|\beta^{L\cup S} - \beta^L\|_2^2 \leq l(\beta^L) - l(\beta^{L\cup S}) + \langle \nabla l(\beta^L), \beta^{L\cup S} - \beta^L \rangle \tag{5.21}$$

We can rearrange and use the fact that $l(.)$ is monotone for increasing supports to obtain

$$0 \leq l(\beta^{L\cup S}) - l(\beta^L) \leq \langle \nabla l(\beta^L), \beta^{L\cup S} - \beta^L \rangle - \frac{m_{\overline{k}}}{2}\|\beta^{L\cup S} - \beta^L\|_2^2 \tag{5.22}$$

$$\leq \max_{\substack{v_{L\cup S^c}=0 \\ v\in C}} \langle \nabla l(\beta^L), v - \beta^L \rangle - \frac{m_{\overline{k}}}{2}\|v - \beta^L\|_2^2 \tag{5.23}$$

$$\leq \max_{v_{L\cup S^c}=0} \langle \nabla l(\beta^L), v - \beta^L \rangle - \frac{m_{\overline{k}}}{2}\|v - \beta^L\|_2^2 \tag{5.24}$$

Where the final inequality is due to the fact that the optimal maximum value of a function subject to some constraint is less than the optimal maximum value of the function not subject to any constraints. Taking the derivative, we find the optimal value of $v = \beta^L + \frac{1}{m_{\overline{k}}}\nabla l(\beta^L)$, which yields a final bound of

$$0 \leq l(\beta^{L\cup S}) - l(\beta^L) \leq \frac{1}{2m_{\overline{k}}}\|\nabla l(\beta^L)\|_2^2 \tag{5.25}$$

Now consider a single $j \in S$. We know $l(\beta^{L\cup\{j\}} \geq l(y_j)$ where $y_j = (1-\alpha_j)\beta^L + (\alpha_j)\beta_j^X$ for all $0 \leq \alpha_j \leq 1$ and any set X. Combining with the smoothness parameter yields

$$l(\beta^{L\cup\{j\}}) - l(\beta^L) \geq l((1-\alpha_j)\beta^L + (\alpha_j)\beta_j^X) - l(\beta^L) \tag{5.26}$$

$$\geq (\alpha_j)\langle \nabla l(\beta^L), \beta_j^X - \beta^L\rangle - \frac{M_{L+1}}{2}(\alpha_j)^2\|\beta_j^X - \beta^L\|_2^2 \tag{5.27}$$

Taking the derivative with respect to α_j yields

$$\alpha_j = \frac{\langle \nabla l(\beta^L), \beta_j^X - \beta^L\rangle}{M_{L+1}\|\beta_j^X - \beta^L\|_2^2} \tag{5.28}$$

Substituting this optimal value of α_j and summing over all $j \in S$ yields

$$l(\beta^{L\cup\{j\}}) - l(\beta^L) \geq \frac{\left(\langle \nabla l(\beta^L), \beta_j^X - \beta^L\rangle\right)^2}{2M_{L+1}\|\beta_j^X - \beta^L\|_2^2} \tag{5.29}$$

$$\implies \sum_{j\in S} l(\beta^{L\cup\{j\}}) - l(\beta^L) \geq \frac{1}{2M_{L+1}}\sum_{j\in S}\left(\nabla l(\beta^L)_j\right)^2 = \frac{1}{2M_{L+1}}\|\nabla l(\beta^L)_S\|_2^2 \tag{5.30}$$

Which when combined with the upper bound on the numerator, gives the desired result. It now remains to show there exists some set X, such that $0 \leq \alpha_j \leq 1$.

Consider the case when $a_j < 0$ for some X such that $L \subset X$. The denominator is strictly positive so $\alpha_j < 0 \implies \langle \nabla l(\beta^L), \beta_j^X - \beta^L\rangle < 0$ for the given X. But this implies for some small positive t, $l((t)\beta_j^X + (1-t)\beta^L) \leq l(\beta^L)$ for all X, but this violates the monotonicity of l, since we have that $l(A) > l(B)$ for $A \subset B$. Thus for any X such that $L \subset X$, $\alpha_j \geq 0$.

We now attempt to upper bound α_j.

$$\alpha_j = \frac{\langle \nabla l(\beta^L), \beta_j^X - \beta^L\rangle}{M_{L+1}\|\beta_j^X - \beta^L\|_2^2} \leq \frac{\max\left\{\|\nabla l(\beta^L)\|_2^2, \|\beta_j^X - \beta_L\|_2^2\right\}}{M_{L+1}\|\beta_j^X - \beta^L\|_2^2} \tag{5.31}$$

If we have that $\|\nabla l(\beta^L)\|_2^2 \leq \|\beta_j^X - \beta_L\|_2^2$, then this implies $\alpha_j \leq \frac{1}{M_{L+1}} \leq 1$ for $M_{L+1} \geq 1$. Now consider the other case, where we have that

$$\alpha_j \leq \frac{\|\nabla l(\beta^L)\|_2^2}{M_{L+1}\|\beta_j^X - \beta^L\|_2^2} \tag{5.32}$$

If there exists some X such that $\frac{\|\nabla l(\beta^L)\|_2^2}{M_{L+1}\|\beta_j^X - \beta^L\|} \leq 1$ then, we have that α_j is also bounded by 1. Now say there exists no such X. We then have

$$\frac{\|\nabla l(\beta^L)\|_2^2}{M_{L+1}\|\beta_j^X - \beta^L\|_2^2} > 1 \implies \frac{\|\nabla l(\beta^L)\|_2^2}{M_{L+1}} > \|\beta_j^X - \beta^L\|_2^2 \ \forall X \subseteq 2^{[p]} \tag{5.33}$$

But then this would violate the Incremental Update Condition (5.3), and thus there some exist some set X such that $\frac{\|\nabla l(\beta^L)\|_2^2}{M_{L+1}\|\beta_j^X - \beta^L\|} \leq 1$, and thus in this case a_j is also bounded by 1.

5.3 PROOF OF CONVERGENCE

Now that we have shown that under certain conditions on the constraint set C, we can use a greedy selection to approximate the optimal support set, we must now prove that this iterative algorithm converges. To do this, we first need to prove some properties of the gradient, which we can derive from the Restricted Strong Convexity condition.

Lemma 5.3 (Monotonicity of gradient for restricted strong convexity)**.** *If f is a function that satisfies the restricted strong convexity condition with parameter μ and domain Ω, then we have that*

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \mu \|x - y\|^2 \ \forall x, y \in \Omega \tag{5.34}$$

Proof (5.4): By the condition of restricted strong convexity we have that

$$f(y) \geq f(x) + \langle \nabla f(x), x - y \rangle + \frac{\mu}{2}\|x - y\|^2 \tag{5.35}$$

$$f(x) \geq f(y) + \langle \nabla f(y), y - x \rangle + \frac{\mu}{2}\|y - x\|^2 \tag{5.36}$$

for all $x, y \in \Omega$. Adding the inequalities together yields the final result that $\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \mu \|x - y\|^2 \ \forall x, y \in \Omega$.

Lemma 5.4. *Let f be a function satisfying the restricted strong convexity condition with parameter μ. Let $x^* = \arg\min_{x \in \Omega} f(x)$ and denote the upper bound of the gradient of f by G. Then for any $y \in \Omega$, we have*

$$-\frac{1}{2\mu}G^2 \leq \nabla f(x^*)^T (y - x^*) \tag{5.37}$$

By the result of [19], we have that restricted strong convexity implies the Polyak- Lojasiewicz (PL) inequality, so we have that

$$\frac{1}{2}\|\nabla f(x)\|^2 \geq \mu(f(x) - f(x^*)) \tag{5.38}$$

Now define $\phi_x(z) = f(z) - \nabla f(x)^T z$. Note that

$$(\nabla\phi_x(z_1) - \nabla\phi_x(z_2))(z_1 - z_2) = (\nabla f(z_1) - \nabla f(z_2))(z_1 - z_2) \geq \mu\|z_1 - z_2\|^2 \tag{5.39}$$

Which implies that $\phi_x(z)$ satisfies the restricted strong convexity condition on domain Ω, by the equivalent characterization of the monotonicity of the gradient. We can then apply the PL inequality to $\phi_x(z)$, which yields

$$\phi_x(x^*) = f(x) - \langle\nabla f(x), x\rangle \geq \phi_x(y) - \frac{1}{2\mu}\|\nabla\phi_x(y)\| \tag{5.40}$$

$$= f(y) - \langle\nabla f(x), y\rangle - \frac{1}{2\mu}\|\nabla f(y) - \nabla f(x)\| \tag{5.41}$$

Which after rearrangement yields

$$f(y) \leq f(x) + \langle\nabla f(x), y - x\rangle + \frac{1}{2\mu}\|\nabla f(y) - \nabla f(x)\|^2 \ \ \forall x, y \in \Omega \tag{5.42}$$

Setting $x = x^*$, upper bounding the gradient by G, and using the fact that $f(x) - f(x^*) \geq 0$, we have

$$0 \leq f(x) - f(x^*) \leq \langle\nabla f(x), y - x\rangle + \frac{1}{2\mu}G^2 \tag{5.43}$$

which implies

$$-\frac{1}{2\mu}G^2 \leq \langle\nabla f(x^*), y - x^*\rangle \ \ \forall y \in \Omega \tag{5.44}$$

Now that we have shown the preliminary proofs regarding the properties of the gradient, we can now begin to present the proof of convergence of the algorithm. The next lemma is a general lemma regarding approximate solutions to optimization problems of the form $x^* = \arg\min_{x\in C\cap S}\{L(x) + B_\psi(x, x_0)\}$. Note that if we replace $L(x)$ with $\langle x, \nabla f(x)\rangle$, then this becomes our iterative subproblem,

Lemma 5.5 (Approximate Bregman Projection)**.** *Let L be a function satisfying the restricted strong sparse convexity conditions (IHT), and define x^* as*

$$x^* = \arg\min_{x \in C \cap S} \{L(x) + B_\psi(x, x_0)\} \tag{5.45}$$

Denote x_k^* as the approximate solution to the above optimization problem, computed via greedy support selection. Let $(1+\phi)$ be the approximation factor of approximate solution. Then for all $y \in C \cap S$ we have

$$(1+\phi)(L(y) + B_\psi(y, x_0)) \geq L(x_k^*) + B_\psi(x_k^*, x_0) + B_\psi(y, x^*) - \frac{1}{2\mu}G^2 \tag{5.46}$$

Proof (5.5): Since x^* is the minimizer of $L(x) + B_\psi(x, x_0)$ over $C \cap S$, which satisfies the restricted strong convexity condition over a sparse domain Ω_k, for $d \in \partial(L(x) + B_\psi(x, x_0))$, we have that

$$\langle d, x - x^* \rangle \geq -\frac{1}{2\mu}G^2 \quad \forall x \in C \cap S \tag{5.47}$$

Note that $d = g + \nabla\psi(x^*) - \nabla\psi(x_0)$ for $g \in \partial L(x)$ which implies there must exist a subgradient $g \in \partial L(x^*)$ such that

$$\langle g + \nabla\psi(x^*) - \nabla\psi(x_0), x - x^* \rangle \geq -\frac{1}{2\mu}G^2 \quad \forall x \in C \cap S \tag{5.48}$$

Using the subgradient property we have

$$L(y) \geq L(x^*) + \langle g, y - x^* \rangle \tag{5.49}$$

$$\geq L(x^*) + \langle \nabla\psi(x_0) - \nabla\psi(x^*), y - x^* \rangle - \frac{1}{2\mu}G^2 \tag{5.50}$$

$$= L(x^*) - \langle \nabla\psi(x_0), x^* - x_0 \rangle + \psi(x^*) - \psi(x_0) \tag{5.51}$$

$$+ \langle \nabla\psi(x_0), y - x_0 \rangle - \psi(y) + \psi(x_0) \tag{5.52}$$

$$- \langle \nabla\psi(x^*), y - x^* \rangle + \psi(y) - \psi(x^*) \tag{5.53}$$

$$= L(x^*) + B_\psi(x^*, x_0) - B_\psi(y, x_0) + B_\psi(y, x^*) - \frac{1}{2\mu}G^2 \tag{5.54}$$

which after rearrangement yields

$$L(x^*) + B_\psi(y, x_0) \geq L(x^*) + B_\psi(x^*, x_0) + B_\psi(y, x^*) - \frac{1}{2\mu}G^2 \quad \forall y \in C \cap S \tag{5.55}$$

Note that our approximation algorithm yields an approximate optimal solution for the value of the function $L(x) + B_\psi(x, x_0)$. We therefore have

$$(1+\phi)(L(x^*)+B_\psi(x^*,x_0)) \geq L(x_k^*)+B_\psi(x_k^*,x_0) \tag{5.56}$$

Substituting x^* with x_k^* yields

$$L(x_k^*)+B_\psi(x_k^*,x_0)+B_\psi(y,x^*)-\frac{1}{2\mu}G^2 \tag{5.57}$$

$$\leq (1+\phi)(L(x^*)+B_\psi(x^*,x_0))+B_\psi(y,x^*)-\frac{1}{2\mu}G^2 \tag{5.58}$$

$$\leq (1+\phi)(L(x^*)+B_\psi(x^*,x_0)+B_\psi(y,x^*)-\frac{1}{2\mu}G^2) \tag{5.59}$$

$$\leq (1+\phi)(L(y)+B_\psi(y,x_0)) \tag{5.60}$$

We now present the final convergence result for the algorithm.

Theorem 5.3 (Convergence)**.** *Let f be a function which is satisfies the RSS and RSC properties with parameters (ω,μ) respectively on domain $D=\{x,y\,:\,\|x\|_0\leq k\,,\|y\|_0\leq k\,,\|x-y\|_0\leq k\}$, with respect to some norm $\|.\|$. Let ψ be a function which is at least ω strongly convex, and let B_ψ be the induced Bregman divergence. Say we run the algorithm for T iterations, let x^* be the optimal solution, let x_i be the i-th iterate and let $\epsilon_i=f(x_i)-f(x^*)$. The convergence rate of the algorithm can then be written as*

$$\min_{i\in T/2\ldots T} e_i \leq \frac{2RG^2}{\sqrt{\mu T}} \tag{5.61}$$

Where $B_\psi(x^*,x_1)\leq R$ and $\|\nabla f(.)\|_*\leq G$

Proof (5.6): Rearranging the result of the previous lemma and considering $L(x)=\langle\nabla f(x),x-x_k\rangle$ yields

$$B_\psi(x^*,x_{k+1}) \leq (1+\phi)(B_\psi(x^*,x_k))+(1+\phi)(\alpha_k)\langle\nabla f(x_k),x^*-x_k\rangle \tag{5.62}$$

$$+\alpha_k\langle\nabla f(x_k),x_k-x_{k+1}\rangle-B_\psi(x_{k+1},x_k)+\frac{1}{2\mu}G^2 \tag{5.63}$$

$$\leq (1+\phi)(B_\psi(x^*,x_k))+(1+\phi)(\alpha_k)(f(x_k)-f(x^*)) \tag{5.64}$$

$$+\alpha_k\langle\nabla f(x_k),x_k-x_{k+1}\rangle-\frac{\mu}{2}\|x_k-x_{k+1}\|+\frac{1}{2\mu}G^2 \tag{5.65}$$

$$\leq (1+\phi)(B_\psi(x^*,x_k))+(1+\phi)(\alpha_k)(f(x_k)-f(x^*)) \tag{5.66}$$

$$+\alpha_k\|\nabla f(x_k)\|_*\|x_k-x_{k+1}\|-\frac{\mu}{2}\|x_k-x_{k+1}\|+\frac{1}{2\mu}G^2 \tag{5.67}$$

$$\leq (1+\phi)B_\psi(x^*,x_k)+(1+\phi)(\alpha_k)(f(x_k)-f(x^*))+\frac{(\alpha_k^2+1)}{2\mu}G^4 \tag{5.68}$$

Where the second inequality follows from the definition of convexity and the quadratic lower bound on Bregman Divergences implied by strong convexity, the third inequality follows from an application of the Cauchy-Schwartz inequality, and the final inequality follows by bounding $\|\nabla f(x_k)\|$ by G.

Now assume we run the algorithm for T iterations, and without loss of generality, assume T is as even integer. Telescoping from $\frac{T}{2}$ to T yields and rearranging yields

$$(1+\phi)(\alpha_i)\sum_{i=T/2}^{T}(f(x_i)-f(x^*)) \leq (1+\phi)R^2 + \frac{G^4}{2\mu}\sum_{i=T/2}^{T}\alpha_i^2 + 1 \tag{5.69}$$

Let $\epsilon_i = f(x_i) - f(x^*)$. We then have

$$\min_{i\in T/2\ldots T}\epsilon_i \leq \frac{(1+\phi)R^2 + \frac{G^4}{2\mu}\sum_{i=T/2}^{T}(\alpha_i^2+1)}{(1+\phi)(\sum_{i=T/2}^{T}\alpha_i)} \tag{5.70}$$

Note that α_i is the step size during the i-th iteration, so we can set the value of α_i to optimize the convergence rate. Let

$$\alpha_i = \frac{R}{G^2\sqrt{i}(1+\phi)} \tag{5.71}$$

which simplifies the earlier expression as follows

$$\frac{G^2}{R}\times\frac{(1+\phi)^2R^2 + \frac{G^4}{2\mu}\sum_{i=T/2}^{T}\left(\frac{R}{G^2\sqrt{k}(1+\phi)}\right)^2 + O(1)}{2\sum_{i=T/2}^{T}\frac{1}{\sqrt{i}}} \tag{5.72}$$

$$= \frac{G^2R}{\sqrt{\mu}}\times\frac{1+\sum_{i=T/2}^{T}\frac{1}{i}}{2\sum_{i=T/2}^{T}\frac{1}{\sqrt{i}}} \leq \frac{2RG^2}{\sqrt{\mu T}} \tag{5.73}$$

where we use the fact that $\log T - \log(\frac{T}{2}-1) \approx \log 2$. This gives an overall convergence rate of

$$\min_{i\in T/2\ldots T}\epsilon_i \leq \frac{2RG^2}{\sqrt{\mu T}} \tag{5.74}$$

which is a factor of G worse than the standard convergence of the Mirror Descent algorithm. Note that the step size is inversely proportional to the approximation quality, so the worse the approximation bound, the slower the algorithm will progress.

CHAPTER 6: CONCLUSIONS

In this thesis, we developed a novel algorithm to solve optimization problems of the form

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad x \in C \cap S_k \end{aligned} \tag{6.1}$$

where C is a convex set, and S_k is a sparsity constraint. We started with an overview of the Iterative Hard Thresholding algorithm, which solves optimization problems of the form

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad x \in S_k \end{aligned} \tag{6.2}$$

and then discussed the celebrated Mirror Descent algorithm, which solves optimization problems of the form

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad x \in C \end{aligned} \tag{6.3}$$

We then discussed previous attempts at modifying Mirror Descent for sparse optimization via regularization methods, and finally we presented a novel algorithm which modifies Mirror Descent to consider the sparsity constraints directly. We showed that although the general problem of projecting onto a sparsity constraint under a Bregman divergence is computationally hard, an approximate projection suffices for iterative convergence. Specifically, we proved that under certain geometrical conditions on the convexity constraint, the result of Elenberg et.al [18] can be applied to bound the submodularity ratio of the subproblem objective function, thus allowing us to bound the approximation guarantee of a greedy support selection algorithm. Future work in this direction will include implementations of the algorithm, and empirical results on regarding the speed of convergence. Hopefully this work has provided insight into the inner working of some of the aforementioned algorithms, and helped the reader develop an understanding of the novel proposed algorithm for sparse Mirror Descent, which will hopefully inspire new ways of thinking of algorithms for sparse optimization.

REFERENCES

[1] P. Jain, A. Tewari, and P. Kar, "On iterative hard thresholding methods for high-dimensional m-estimation," *CoRR*, vol. abs/1410.5137, 2014. [Online]. Available: http://arxiv.org/abs/1410.5137

[2] E. Candes and T. Tao, "Decoding by linear programming," 2005. [Online]. Available: https://arxiv.org/abs/math/0502327

[3] A. Argyriou, R. Foygel, and N. Srebro, "Sparse prediction with the k-support norm," 2012. [Online]. Available: https://arxiv.org/abs/1204.5043

[4] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and computational harmonic analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[5] P. Jain and P. Kar, "Non-convex optimization for machine learning," *Foundations and Trends® in Machine Learning*, vol. 10, no. 3-4, pp. 142–336, 2017. [Online]. Available: https://doi.org/10.15612F2200000058

[6] E. R. Elenberg, R. Khanna, A. G. Dimakis, and S. Negahban, "Restricted strong convexity implies weak submodularity," 2016. [Online]. Available: https://arxiv.org/abs/1612.00804

[7] G. Nemhauser, L. Wolsey, and M. Fisher, "An analysis of approximations for maximizing submodular set functions—i," *Mathematical Programming*, vol. 14, pp. 265–294, 12 1978.

[8] O. O. Koyejo, R. Khanna, J. Ghosh, and R. Poldrack, "On prior distributions and approximate inference for structured variables," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/74071a673307ca7459bcf75fbd024e09-Paper.pdf

[9] B. Kim, R. Khanna, and O. O. Koyejo, "Examples are not enough, learn to criticize! criticism for interpretability," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016. [Online]. Available: https://proceedings.neurips.cc/paper/2016/file/5680522b8e2bb01943234bce7bf84534-Paper.pdf

[10] T. Blumensath, M. Yaghoobi, and M. E. Davies, "Iterative hard thresholding and l0 regularisation," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, vol. 3, 2007, pp. III–877–III–880.

[11] P. Jain, N. Rao, and I. Dhillon, "Structured sparse regression via greedy hard-thresholding," 2016. [Online]. Available: https://arxiv.org/abs/1602.06042

[12] S. Bubeck, "Convex optimization: Algorithms and complexity," 2014. [Online]. Available: https://arxiv.org/abs/1405.4980

[13] A. Beck and M. Teboulle, "Mirror descent and nonlinear projected subgradient methods for convex optimization," *Operations Research Letters*, vol. 31, no. 3, pp. 167–175, 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167637702002316

[14] S. Shalev-Shwartz and A. Tewari, "Stochastic methods for ¡i¿l¡/i¿¡sub¿1¡/sub¿-regularized loss minimization," *J. Mach. Learn. Res.*, vol. 12, no. null, p. 1865–1892, jul 2011.

[15] J. Langford, L. Li, and T. Zhang, "Sparse online learning via truncated gradient," 2008. [Online]. Available: https://arxiv.org/abs/0806.4686

[16] D. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

[17] A. Das and D. Kempe, "Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection," 2011. [Online]. Available: https://arxiv.org/abs/1102.3975

[18] E. R. Elenberg, R. Khanna, A. G. Dimakis, and S. Negahban, "Restricted strong convexity implies weak submodularity," 2017.

[19] H. Karimi, J. Nutini, and M. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition," *CoRR*, vol. abs/1608.04636, 2016. [Online]. Available: http://arxiv.org/abs/1608.04636