Scalable Greedy Feature Selection via Weak Submodularity

Rajiv Khanna UT Austin
Ethan R. Elenberg UT Austin
Alexandros G. Dimakis UT Austin
Sahand Neghaban Yale University
Joydeep Ghosh UT Austin

Abstract

Greedy algorithms are widely used for problems in machine learning such as feature selection and set function optimization. Unfortunately, for large datasets, the running time of even greedy algorithms can be quite high. This is because for each greedy step we need to refit a model or calculate a function using the previously selected choices and the new candidate.

Two algorithms that are faster approximations to the greedy forward selection were introduced recently [Mirzasoleiman et al., 2013, 2015]. They achieve better performance by exploiting distributed computation and stochastic evaluation respectively. Both algorithms have provable performance guarantees for submodular functions.

In this paper we show that divergent from previously held opinion, submodularity is not required to obtain approximation guarantees for these two algorithms. Specifically, we show that a generalized concept of weak submodularity suffices to give multiplicative approximation guarantees. Our result extends the applicability of these algorithms to a larger class of functions. Furthermore, we show that a bounded submodularity ratio can be used to provide data dependent bounds that can sometimes be tighter also for submodular functions. We empirically validate our work by showing superior performance of fast greedy approximations versus several established baselines on artificial and real datasets.

1 Introduction

Consider the problem of sparse linear regression:

$$\min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \text{ s.t. } \|\boldsymbol{\beta}\|_0 \leq k, \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{n \times d}$ is the design matrix (also called feature matrix) for n samples and d features and $\mathbf{y} \in \mathbb{R}^n$ is the corresponding vector of n responses or observations. We assume without loss of generality that the columns of the matrix $\mathbf{X}$ are normalized to 1, and the response vector is also normalized.

Given a subset S of features (denoted by $\mathbf{X}_\mathsf{S}$), it is easy to find the best regression coefficients by projecting $\mathbf{y}$ on the span of $\mathbf{X}_\mathsf{S}$. The R^2 statistic (*coefficient of determination*) measures the proportion of the variance explained by this subset. Sparse regression can be therefore seen as maximizing a *set function* $R^2(\mathsf{S})$: for any given set of columns S , $R^2(\mathsf{S})$ measures how well these columns explain the observations $\mathbf{y}$.

This set function $R^2(\mathsf{S})$ is *monotone* increasing: including more features can only increase the explained variance[1]. Solving sparsity constrained maximization of R^2 would involve searching over all subsets of size up to k and selecting the one that maximizes R^2. This combinatorial optimization problem is unfortunately NP-Hard [Das and Kempe, 2011] . A widely used approach is to use greedy forward selection: select one feature at a time, greedily choosing the one that maximizes R^2 in the next step. Orthogonal Matching Pursuit and variations [Tropp, 2004] [Needell and Tropp, 2008] can also be seen as approximate accelerated greedy forward selection algorithms that avoid re-fitting the model for each candidate vector at each step.

When maximizing set functions using the greedy algorithm it is natural to consider the framework of submodularity. In a classical result, Nemhauser et al. [1978] show that for submodular monotone functions, the greedy k-sparse solution is within $(1 - \frac{1}{e})$ of the optimal k-sparse solution, *i.e.*, greedy gives a constant multiplicative factor approximation guarantee. The concept of submodularity has led to several effective greedy solutions to problems such as sparse prediction [Koyejo et al., 2014], sparse factor analysis [Khanna et al., 2015], model interpretation [Kim et al., 2016], etc.

Unfortunately, it is easy to construct counterexamples to show that $R^2(\mathsf{S})$ is not submodular [Das and Kempe, 2011, Elenberg et al., 2016]. In their breakthrough paper Das and Kempe [2011] show that if the design matrix $\mathbf{X}$ sat-

Proceedings of the 20^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54.

[1] However, *adjusted* R^2 is not monotonic.

isfies the Restricted Isometry Property (RIP), then the set function $R^2(\mathsf{S})$ satisfies a weakened form of submodularity. This weak form of submodularity is obtained by bounding a quantity called a submodularity ratio γ defined subsequently. The authors further showed that a bounded submodularity ratio γ is sufficient for Nemhauser's proof to go through with a relaxed approximation constant (that depends on γ). Therefore, even weak submodularity implies a constant factor approximation for greedy and RIP implies weak submodularity.

The weak submodularity framework was recently extended beyond linear regression by Elenberg et al. [2016] to concave functions (for example, likelihood functions of generalized linear models). This generalization of the RIP condition is called Restricted Strong Convexity (RSC) and the general result is that RSC implies weak submodularity for the set function obtained from the likelihood of the model restricted to subsets of features. This shows that greedy feature selection can obtain constant factor approximation guarantees in a very general setting, similar to results obtained by Lasso but without further statistical assumptions.

Running the greedy algorithm can be computationally expensive for large datasets. This is because for each greedy step we need to refit the model using the previously selected choices and the new candidate. This has led to the development of faster variants. For example, DISTRIBUTEDGREEDY [Mirzasoleiman et al., 2013] distributes the computational effort across available machines, and STOCHASTICGREEDY [Mirzasoleiman et al., 2015] exploits a stochastic greedy step. Both algorithms have provable performance guarantees for submodular functions.

In this paper, we show that submodularity is not required to obtain approximation guarantees for these two algorithms and that a bounded submodularity ratio suffices. This extends the scope of these algorithms to significantly larger class of functions like sparse linear regression with RIP design matrices. Furthermore, as we shall discuss, submodularity ratio can be used to provide data dependent bounds. This implies that one can sometimes obtain tighter guarantees also for submodular functions.

Our contributions are as follows: (1) We analyze and obtain approximation guarantees for DISTRIBUTEDGREEDY and STOCHASTICGREEDY using the submodularity ratio extending the scope of their application to non-submodular functions,
(2) We show that the submodularity ratio can give tighter data dependent bounds even for submodular functions,
(3) We derive explicit bounds for both DISTRIBUTEDGREEDY and STOCHASTICGREEDY for the special case of sparse linear regression.
(4) We derive bounds for sparse support selection for general concave functions.
(5) We also present empirical evaluations of these algorithms vs several established baselines on simulated and real world datasets.

Related Work Das and Kempe [2011] defined submodularity ratio, and showed that for any function that has its submodularity ratio bounded away from 0, one can provide appropriate greedy guarantees. Elenberg et al. [2016] used the concept to derive a new relationship between submodularity and convexity, specifically stating that the Restricted Strong Concavity can be used to bound the submodularity ratio. This results in providing bounds for greedy support selection for general concave functions.

Another notion of approximate additive submodularity was explored by Krause and Cevher [2010] for the problem of dictionary selection. This was, however, superceded by [Das and Kempe, 2011] who showed that submodularity ratio provides tighter approximation bounds. Horel and Singer [2016] consider another generalization from submodular functions – ε-approximate submodular functions which are functions within ε of some submodular function, and provide approximation guarantees for greedy maximization.

The DISTRIBUTEDGREEDY algorithm was introduced by Mirzasoleiman et al. [2013]. They provide deterministic bounds for any arbitrary distribution of data onto the individual machines. Barbosa et al. [2015] showed that for sparsity constraints, and under the assumption that the data is split uniformly at random to all the machines, one can obtain a $\frac{1}{2}(1 - 1/e)$ guarantee in expectation. Kumar et al. [2013] also provide distributed algorithms for maximizing a monotone submodular function subject to a sparsity constraint. They extend the Threshold Greedy algorithm of Gupta et al. [2010] by augmenting it with a sample and prune strategy. It runs the Threshold Greedy algorithm on a subset of data to obtain a candidate solution. The latter is then used to prune the remaining data to reduce its size. This process is repeated a constant number of times, and the algorithm provides a constant factor guarantee

Bhaskara et al. [2016] provide approximation guarantees for greedy selection variants for column subset selection. They do not use the submodularity framework, and their results are not directly applicable or useful for other problem settings. Similarly, Farahat et al. [2013] also use greedy column subset selection. However, their focus is not towards obtaining approximation guarantees, but rather on more efficient algorithmic implementation.

2 Background

Notation: We represent vectors as small letter bolds e.g. $\mathbf{u}$. Matrices are represented by capital bolds e.g. $\mathbf{X}, \mathbf{T}$. Matrix or vector transposes are represented by superscript $\mathbf{X}^\top$. Identity matrices of size s are represented by $\mathbf{I}_s$, or simply $\mathbf{I}$ when the dimensions are obvious. $\mathbf{1}(\mathbf{0})$ is a column vector of all ones (zeroes). Sets are represented by sans serif fonts

e.g. S, complement of a set S is S^c. For a vector $\mathbf{u} \in \mathbb{R}^d$, and a set S of support dimensions with $|\mathsf{S}| = k, k \leq d$, $\mathbf{u}_\mathsf{S} \in \mathbb{R}^k$ denotes subvector of $\mathbf{u}$ supported on S. Similarly, for a matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, $\mathbf{X}_\mathsf{S} \in \mathbb{R}^{k\times k}$ denotes the submatrix supported on S. We denote $\{1, 2, \ldots, d\}$ as $[d]$.

Throughout this manuscript, we assume the set function $f(\cdot)$ is monotone. Our goal is to maximize $f(\cdot)$ under a cardinality constraint :

$$\max_{|\mathsf{S}|\leq k} f(\mathsf{S}). \tag{2}$$

We begin by defining the submodularity ratio of a set function $f(\cdot)$.

Definition 1 (Submodularity Ratio [Das and Kempe, 2011])**.** *Let $\mathsf{S}, \mathsf{L} \subset [d]$ be two disjoint sets, and $f : [d] \to \mathbb{R}$. The submodularity ratio for S with respect to L is given by*

$$\gamma_{\mathsf{L},\mathsf{S}} := \frac{\sum_{j\in\mathsf{S}} [f(\mathsf{L}\cup\{j\}) - f(\mathsf{L})]}{f(\mathsf{L}\cup\mathsf{S}) - f(\mathsf{L})}. \tag{3}$$

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

$$\gamma_{\mathsf{U},k} := \min_{\substack{\mathsf{L},\mathsf{S}:\mathsf{L}\cap\mathsf{S}=\emptyset,\\ \mathsf{L}\subseteq\mathsf{U},|\mathsf{S}|\leq k}} \gamma_{\mathsf{L},\mathsf{S}}. \tag{4}$$

It is straightforward to show that f is submodular if and only if $\gamma_{\mathsf{L},\mathsf{S}} \geq 1$ for all sets L and S. Generalizing to the functions with $0 < \gamma_{\mathsf{L},\mathsf{S}} \leq 1$ provides a notion of *weak submodularity* [Elenberg et al., 2016]. For weakly submodular functions, even though the function may not be submodular, it still provides approximation guarantees for the greedy algorithm. We use the submodularity ratio to provide new bounds for DISTRIBUTEDGREEDY and STOCHASTICGREEDY, thereby generalizing these algorithms to non-submodular functions.

2.1 Greedy Selection

We briefly go over the classic greedy algorithm for subset selection. A greedy approach to optimizing a set function is myopic – the algorithm chooses the element from the available choices that gives the largest *incremental* gain for the set of choices previously made. The algorithm is illustrated in Algorithm 1. The algorithm makes k *outer* iterations, where k is the desired sparsity. Each iteration is a full pass over the remaining candidate choices, wherein the marginal gain is calculated for each remaining candidate choice. Thus the greedy algorithm has the computational complexity of $O(dk)$ calls to the function evaluation oracle.

For a large d, the linear scaling of the greedy algorithm for a fixed k may be prohibitive. As such, algorithms that scale sublinearly are useful for truly large scale selections. The DISTRIBUTEDGREEDY algorithm, for example, achieves this sublinear scaling by making use of multiple machines. The data is split uniformly at random to l machines. Each machine then performs its own independent greedy selection (Algorithm 1), and outputs a k sized solution. All of the greedy solutions are collated by a central machine, which performs another round of the greedy selections to output the final solution. The algorithm is illustrated in Algorithm 2, and is analyzed in Section 3. It has a computational complexity of $O(dk/l)$ The algorithm is easy to implement in parallel or within a distributed computing framework *e.g.* MapReduce.

Algorithm 1 GREEDY(S, k)

1: Input: sparsity k, available choices S
2: $\mathsf{A}_0 = \emptyset$
3: **for** $i \in 0 \ldots (k-1)$ **do**
4: $\quad s = \arg\max_{j\in\mathsf{S}\setminus\mathsf{A}_i} f(\mathsf{A}_i \cup \{j\}) - f(\mathsf{A}_i)$
5: $\quad \mathsf{A}_{i+1} = \mathsf{A}_i \cup \{s\}$
6: **end for**
7: return A_k

Algorithm 2 DISTRIBUTEDGREEDY(l, k,$\{\mathsf{A}_j\}$)

1: Input: sparsity k, number of parallel solvers l, partition $\{\mathsf{A}_j\}$ of the set of available choices A
2: $\mathsf{G}_i \leftarrow$ GREEDY(A_j, k) $\forall j \in [l]$
3: $\mathsf{G} \leftarrow$ GREEDY($\cup_j \mathsf{G}_j$, k)
4: $\mathsf{G}_{\max} \leftarrow \arg\max_{\mathsf{G}_j} f(\mathsf{G}_j)$
5: return $\arg\max f(\mathsf{G}), f(\mathsf{G}_{\max})$

An alternative to distributing the data, say when several machines are not available, is to perform the greedy selection *stochastically*. The STOCHASTICGREEDY algorithm for submodular functions was introduced by Mirzasoleiman et al. [2015]. At any given iteration $i \in [k]$, instead of performing a function evaluation for each of the remaining $(d-i)$ candidates, a subset of a fixed size $C = \lceil \frac{d \log 1/\delta}{k} \rceil$ (where δ is a pre-specified hyperparameter) is uniformly sampled from the available $(d-i)$ choices using the subroutine SUBSAMPLE, and the function evaluation is made on those subsampled choices as if they were the only available candidates. This speeds up the greedy algorithm to $O(Ck)$ function evaluations. The algorithm is presented in Algorithm 3, and its approximation bounds are discussed in Section 4.

Finally, we discuss a property of the greedy algorithm, which is fundamental to the analysis of the DISTRIBUTEDGREEDY algorithm. The greedy algorithm belongs to a larger class of algorithms called 1-nice algorithms [Mirrokni and Zadimoghaddam, 2015]. The following result allows us to remove or add unselected items from the choice set that is accessible to the algorithm.

Lemma 1. *[Mirrokni and Zadimoghaddam, 2015] Say $|\mathsf{S}| > k$, and let GREEDY$(\mathsf{S}, k) \subset \mathsf{S}$ be the k-sized set*

Algorithm 3 STOCHASTICGREEDY(S, k, δ)

1: Input: sparsity k, available choices S, subsampling parameter δ
2: $\mathsf{A}_0 = \emptyset$
3: **for** $i \in 0 \dots (k-1)$ **do**
4: $\quad \mathsf{S}_\delta \leftarrow$ SUBSAMPLE($\mathsf{S} \backslash \mathsf{A}_i$, δ, k)
5: $\quad s = \arg\max_{j \in \mathsf{S}_\delta} f(\mathsf{A}_i \cup \{j\}) - f(\mathsf{A}_i)$
6: $\quad \mathsf{A}_{i+1} = \mathsf{A}_i \cup \{s\}$
7: **end for**
8: return A_k

returned by Algorithm 1. For any $x \notin \text{GREEDY}(\mathsf{S}, k)$, $\text{GREEDY}(\mathsf{S} \backslash \{x\}, k) = \text{GREEDY}(\mathsf{S}, k)$.

Note that Lemma 1 is a property of the algorithm, and is independent of the function itself. Prior works [Mirrokni and Zadimoghaddam, 2015], [Barbosa et al., 2015] have exploited this property in conjunction with properties of submodular functions to obtain approximation bounds for the distributed algorithms. Our work extends these results to weakly submodular functions. As such, it is easy to see that our results are easily extensible to other *nice* algorithms – including distributed OMP and distributed stochastic greedy – that have closed form bounds for the respective single machine algorithm. For ease of exposition, we focus our discussion on the distributed greedy algorithm.

3 Distributed Greedy

In this section, we obtain approximation bounds for DISTRIBUTEDGREEDY (Algorithm 2). The algorithm returns the best out of $(l+1)$ solutions : the l *local* solutions (steps 2,4), and the final aggregated one (step 3). Our strategy to obtain the approximation bound for the algorithm is as follows. To obtain an overall approximation bound, we obtain individual bounds on each of the solutions in terms of the submodularity ratio (Definition 1) and use the subadditivity ratio (Definition 2) to show that one of the two shall always hold. For approximation bounds on the *local* solutions, we make use of the *niceness* of the GREEDY (Lemma 1). The bound on the aggregated solution is more involved, since it involves tracking the split of the true solution $\mathsf{A}^\star$ across machines. The assumption of partitioning uniformly at random is vital here. This helps us lower bound the greedy gain in expectation by a probabilistic overlap with the true solution. The trick of tracking the split of the true solution across machines is similar to the one that has been used for analysis of submodular functions [Mirrokni and Zadimoghaddam, 2015], [Barbosa et al., 2015], but without the explicit connection to submodularity and subadditivity ratios. As we shall see in Sections 5, 6 elucidating these connections leads to novel bounds for support selection for linear regression and general concave functions.

We next define the subadditivity ratio, which helps us generalize subadditive functions in the way similar to how submodularity ratio generalizes submodular functions.

Definition 2 (Subaddivity ratio)**.** *We define the subadditivity ratio for a set function f w.r.t a set S as:*

$$\nu_{\mathsf{S}} := \min_{\substack{\mathsf{A} \cup \mathsf{B} = \mathsf{S} \\ \mathsf{A} \cap \mathsf{B} = \phi}} \frac{f(\mathsf{A}) + f(\mathsf{B})}{f(\mathsf{S})}.$$

We further define the subadditivity ratio of a function for an integer k, ν_k, which takes a uniform bound over all sets of size k:

$$\nu_k := \min_{\mathsf{S}:|\mathsf{S}|=k} \nu_{\mathsf{S}}.$$

By definition, the function $f(\cdot)$ is subadditive iff $\nu_{\mathsf{S}} \geq 1, \forall \mathsf{S} \subset [d]$. Since submodularaity implies subadditivity (the converse is not always true), if the function $f(\cdot)$ is submodular, $\nu_{\mathsf{S}} \geq 1, \forall \mathsf{S} \subset [d]$.

We next present some notation and few lemmas that lead up to the main result of this section (Theorem 1). Let A be the entire set of available choices. Partition the set A uniformly at random into $\mathsf{A}_1, \dots, \mathsf{A}_l$. Let G_j be the k-sized solution returned by running the greedy algorithm on A_j *i.e.* $\mathsf{G}_j = \text{GREEDY}(\mathsf{A}_j, k)$. Note that each A_j induces a partition onto the optimal k-sized solution A^* as follows:

$$\mathsf{S}_j := \{x \in \mathsf{A}^* : x \in \text{GREEDY}(\mathsf{A}_j \cup x, k)\},$$
$$\mathsf{T}_j := \{x \in \mathsf{A}^* : x \notin \text{GREEDY}(\mathsf{A}_j \cup x, k)\}.$$

Having defined the notation, we start by lower bounding the local solutions in terms of value of the subset of $\mathsf{A}^\star$ that is not selected as part of the respective local solution.

Lemma 2. $f(\mathsf{G}_j) \geq (1 - \exp(-\gamma_{\mathsf{G}_j,k}))f(\mathsf{T}_j)$.

The next lemma is used to lower the bound the value of the aggregated solution (step 4 in Algorithm 2) in terms of the value of the subset $\mathsf{A}^\star$ that is selected as part of the respective local solution.

Lemma 3. $\exists j \in [l]$ *s.t.* $\mathbb{E}[f(\mathsf{G})] \geq \left(1 - \frac{1}{e^{\gamma_{\mathsf{G},k}}}\right) f(\mathsf{S}_j)$.

We are now ready to present our main result about the approximation guarantee for Algorithm 2.

Theorem 1. *Let G_{dg} be the set returned by the distributed greedy (Algorithm 2). Let $\gamma = \min\{\gamma_{\mathsf{G}_i,k}, \gamma_{\mathsf{G},k}\}$. Then,*

$$\mathbb{E}[f(\mathsf{G}_{dg})] \geq \frac{\nu_k}{2} \left(1 - \exp(-\gamma)\right) f(\mathsf{A}^*). \tag{5}$$

Proof. There are l machines, each with its local greedy solution $\mathsf{G}_i, i \in [l]$. In addition, there is the aggregated solution set G. The key idea is to show that atleast one of the $(l+1)$ solutions is *good enough*.

Say $f(\mathsf{T}_i) \geq \frac{\nu_k}{2} f(\mathsf{A}^*)$ for some i, then by Lemma 2, $f(\mathsf{G}_i) \geq \frac{\nu_k}{2}(1 - \exp(-\gamma_{\mathsf{G}_i,k}))f(\mathsf{A}^*)$.

On the other hand, say for all i, $f(\mathsf{T}_i) < \frac{\nu_k}{2} f(\mathsf{A}^*)$, then for all i, $f(\mathsf{S}_i) > \frac{\nu_k}{2} f(\mathsf{A}^*)$. By Lemma 3, the result then follows. □

Theorem 2 generalizes the approximation guarantee of $\frac{1}{2}(1 - \frac{1}{e})$ obtained by Barbosa et al. [2015] for submodular functions. Their analysis uses convexity of the Lovasz extension of submodular functions, and hence can not be trivially extended to weakly submodular functions. In addition to being applicable for a larger class of functions, our result can also provide tighter bounds for specific applications or datasets even for submodular functions, since they are also applicable for submodular functions, and bounding ν_k and γ away from 1 from domain knowledge will give tighter approximations than the generic bound of $\frac{1}{2}(1 - \frac{1}{e})$.

4 Stochastic Greedy

For analysis of Algorithm 3, we show that the subsampling parameter δ governs the tradeoff between the speedup and the loss in approximation quality *vis-a-vis* the classic GREEDY. Before formally providing the approximation bound, we present an auxillary lemma that is key to proving the new approximation bound. The following result is a generalization of Lemma 2 from Mirzasoleiman et al. [2015] for weakly submodular functions.

Lemma 4. *Let $\mathsf{A}, \mathsf{B} \subset [n]$, with $|\mathsf{B}| \leq k$. Consider another set C drawn randomly from $[n] \backslash \mathsf{A}$ with $|\mathsf{C}| = \lceil \frac{n \log 1/\delta}{k} \rceil$. Then,*

$$\mathbb{E}[\max_{v \in \mathsf{C}} f(v \cup \mathsf{A}) - f(\mathsf{A})] \geq \frac{(1-\delta)\gamma_{\mathsf{A},\mathsf{B}\backslash\mathsf{A}}}{k}(f(\mathsf{B}) - f(\mathsf{A})).$$

We are now ready to present our result that shows that stochastic greedy selections (Algorithm 3) can be applied to weakly submodular functions with provable approximation guarantees. All the proofs missing from the main text are presented in the supplement.

Theorem 2. *Let $\mathsf{A}^\star$ be the optimum set of size k, and $\mathsf{A}_i = \{a_1, a_2, \ldots, a_i\}, i \leq k$ be the set built by* STOCHASTICGREEDY *at step i. Then, $\mathbb{E}[f(\mathsf{A}_k)] \geq \left(1 - \frac{1}{e^{\gamma_{\mathsf{A}_k,k}}} - \delta\right) f(\mathsf{A}^*)$.*

Proof. Define $g_i := f(\mathsf{A}_i) - f(\mathsf{A}_{i-1})$. Using Lemma 4 with $\mathsf{B} = \mathsf{A}^\star$ and $\gamma_{\mathsf{A}_{i-1},\mathsf{B}\backslash\mathsf{A}} \geq \gamma_{\mathsf{A}_k,k}$, we get at the i-th step,

$$\begin{aligned}\mathbb{E}[g_i|\mathsf{A}_{i-1} = \mathsf{A}] \geq & \quad \frac{(1-\delta)\gamma_{\mathsf{A}_k,k}}{k}(f(\mathsf{B}) - f(\mathsf{A})) \\ \geq & \quad \frac{(1-\delta)\gamma_{\mathsf{A}_k,k}}{k}(f(\mathsf{A}^\star) - f(\mathsf{A})). \quad (6)\end{aligned}$$

Define $h_{i-1} := \mathbb{E}[f(\mathsf{A}^\star) - f(\mathsf{A}_{i-1})]$, $C := \frac{(1-\delta)\gamma_{\mathsf{A}_k,k}}{k}$. Note that $\mathbb{E}[g_i] = h_{i-1} - h_i$. Taking expectation on both sides over A_i, (6) becomes

$$h_i \leq (1 - C)h_{i-1} \leq (1 - C)^i h_0.$$

Using $h_k = \mathbb{E}[f(\mathsf{A}^\star) - f(\mathsf{A}_k)]$ and $h_0 = f(\mathsf{A}^\star)$ above, along with the fact that $1 + a\delta \geq a^\delta$ for $0 \leq \delta \leq 1$,

$$\begin{aligned}\mathbb{E}[f(\mathsf{A}_k)] \quad &\geq \quad \left(1 - \left(1 - \frac{(1-\delta)\gamma_{\mathsf{A}_k,k}}{k}\right)^k\right) f(\mathsf{A}^*) \\ &\geq \quad \left(1 - \exp\left(-\gamma_{\mathsf{A}_k,k}\,(1-\delta)\right)\right) f(\mathsf{A}^*) \\ &\geq \quad \left(1 - \frac{1}{e^{\gamma_{\mathsf{A}_k,k}}} - \delta\right) f(\mathsf{A}^*). \quad \square\end{aligned}$$

Note that δ is the tradeoff hyperparameter between the speedup achieved by subsampling and the corresponding approximation guarantee. A larger value of δ means the algorithm is faster with weaker guarantees and vice versa. As $\delta \to 0$, we tend towards the bound $(1 - \frac{1}{e^{\gamma_{\mathsf{A}_k,k}}})$ which recovers the bound for weakly submodular functions obtained by Das and Kempe [2011] for the classic greedy selections (Algorithm 1), and also recovers the well known bound of $(1 - \frac{1}{e})$ for submodular functions.

5 Large Scale Sparse Linear Regression

In this section, we derive novel bounds for greedy support selections for linear regression using both Algorithms 2, 3. Recall that $\mathbf{X} \in \mathbb{R}^{n \times d}$ is the *feature matrix*, with n samples and d features, and $\mathbf{y} \in \mathbb{R}^n$ is the vector of n *responses*. We assume, without loss of generality, that the columns of the matrix $\mathbf{X}$ are normalized to 1, and the response vector is also normalized to have norm 1. Let $\mathbf{C} \in \mathbb{R}^{d \times d}$ be the covariance matrix.

We know from standard linear algebra, that for a fixed set of columns $\mathbf{X}_\mathsf{S}$, where $\mathsf{S} \subset [d]$ is the index into columns of $\mathbf{X}$, $\boldsymbol{\beta}^\star_\mathsf{S} = (\mathbf{X}_\mathsf{S}^\top \mathbf{X}_\mathsf{S})^{-1} \mathbf{X}_\mathsf{S}^\top \mathbf{y}$. Minimizing the error in (1) is thus equivalent to maximizing the following set function (modulo a constant $\|\mathbf{y}\|_2^2$):

$$f(\mathsf{S}) := \|\mathbf{P}_\mathsf{S}\mathbf{y}\|_2^2, \quad (7)$$

where $\mathbf{P}_\mathsf{S} := \mathbf{X}_\mathsf{S}(\mathbf{X}_\mathsf{S}^\top \mathbf{X}_\mathsf{S})^{-1}\mathbf{X}_\mathsf{S}^\top$ is the projection matrix for orthogonal projection onto the span of columns of $\mathbf{X}_\mathsf{S}$. $f(\cdot)$ as defined above is also the R^2 statistic for the linear regression problem (1). The respective combinatorial maximization is:

$$\max_{|\mathsf{S}| \leq k} f(\mathsf{S}). \quad (8)$$

The function defined in (8) is *not* submodular. However, submodularity is not required for giving guarantees for

greedy forward selection. A bounded submodularity ratio is a weaker condition that is sufficient to provide such approximation guarantees. Das and Kempe [2011] analyzed the greedy algorithm for (1), and showed that this function was weakly submodular. Our goal is to maximize the R^2 statistic for linear regression using Algorithms 2, 3.

For a positive semidefinite matrix $\mathbf{C}$, $\lambda_{\min}(\mathbf{C}, k)$ and $\lambda_{\min}(\mathbf{C}, k)$ be the largest and smallest k-sparse eigenvalue of $\mathbf{C}$. We make use of the following result from Das and Kempe [2011]:

Lemma 5. *For the R^2 statistic* (7)*, $\gamma_{\mathsf{S},k} \geq \lambda_{\min}(\mathbf{C},\, k+|\mathsf{S}|)$.*

Recall that we need to only bound the submodularity ratio $\gamma_{\mathsf{S}_g,k}$ over the selected greedy set to obtain the approximation bounds (See Theorems 1 and 2). Lemma 5 provides a data dependent union bound for the submodularity ratio in terms of sparse eigenvalue of the covariance matrix. We now provide the corresponding approximation bounds for Algorithm 3 next.

Corollary 1. *Let $\mathsf{A}^\star$ be the optimal support set for sparsity constrained maximization of the R^2 statistic* (8)*. Let A_{sg} be the solution returned by* StochasticGreedy *(S, k, δ). Then,*

$$\mathbb{E}[f(\mathsf{A}_{sg})] \geq \left(1 - \frac{1}{e^{\lambda_{\min}(C,\, 2k)}} - \delta\right) f(\mathsf{A}^*).$$

To obtain bounds for Algorithm 2, we also need to bound the subadditivity ratio (recall Definition 2).

Lemma 6. *For the maximization of the R^2* (7)*, $\nu_{\mathsf{S}} \geq \frac{\lambda_{\min}(\mathbf{C}_{\mathsf{S}})}{\lambda_{\max}(\mathbf{C}_{\mathsf{S}})}$, where $\mathbf{C}_{\mathsf{S}}$ is the submatrix of $\mathbf{C}$ with rows and columns indexed by S.*

We can now provide the bounds for greedy support selection using Algorithm 2 for the linear regression problem (7).

Theorem 3. *Let A_{dg} be the solution returned by the* DistributedGreedy *algorithm, and let $\mathsf{A}^\star$ be the optimal solution for the sparsity constrained maximization of R^2* (8)*. Then,*

$$\begin{aligned} f(\mathsf{A}_{dg}) &\geq \frac{1}{2}\frac{\lambda_{\min}(\mathbf{C}_{\mathsf{A}^\star})}{\lambda_{\max}(\mathbf{C}_{\mathsf{A}^\star})}\left(1 - \exp(-\lambda_{\min}(\mathbf{C},\, 2k))\right) f(\mathsf{A}^*) \\ &\geq \frac{1}{2}\frac{\lambda_{\min}(\mathbf{C}, k)}{\lambda_{\max}(\mathbf{C}, k)}\left(1 - \exp(-\lambda_{\min}(\mathbf{C},\, 2k))\right) f(\mathsf{A}^*). \end{aligned}$$

Proof. Follows from Lemma 6 and by a uniform bound on γ in Theorem 1 as $\gamma \geq \lambda_{\min}(\mathbf{C}, 2k)$ (from Lemma 5). □

6 Support Selection for general functions

In this section, we leverage recent results from connections of convexity to submodularity to provide support selection bounds for Algorithms 2, 3 for general concave functions.

The sparsity constraint problem with a given $k \leq d$ for a concave function $g : \mathbb{R}^d \to \mathbb{R}$ is:

$$\max_{\|\mathbf{x}\|_0 \leq k} g(\mathbf{x}). \tag{9}$$

Similar to the developments in Section 5, we can define an associated set function as:

$$f(\mathsf{S}) := \max_{\text{supp}(\mathbf{x}) \subset \mathsf{S}} g(\mathbf{x}) - g(\mathbf{0}). \tag{10}$$

We recall that the submodular guarantee of $(1 - \frac{1}{e})$ is for *normalized* submodular functions. To extend the notion of normalization to general support selection, we subtract $g(\mathbf{0})$.

To bound the submodularity ratio for $f(\cdot)$ in (10), the concept of strong concavity and smoothness is required. For a function $g : \mathbb{R}^d \to \mathbb{R}$, define $\mathcal{D}_g(\mathbf{x}, \mathbf{y}) := g(\mathbf{y}) - g(\mathbf{x}) - \langle \nabla g(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle$. We say $g(\cdot)$ is m_Ω Restricted Strongly Concave (RSC) and L_Ω Restricted Strongly Smooth (RSM) over a subdomain $\Omega \subset \mathbb{R}^d$ if

$$-\frac{m_\Omega}{2}\|\mathbf{y} - \mathbf{x}\|_2^2 \geq \mathcal{D}_g(\mathbf{x}, \mathbf{y}) \geq -\frac{L_\Omega}{2}\|\mathbf{y} - \mathbf{x}\|_2^2.$$

We make use of the following result that lower bounds the submodularity ratio for $f(\cdot)$:

Lemma 7 (Elenberg et al. [2016])**.** *If the given function $g(\cdot)$ is m-strongly concave on all $|\mathsf{S}| + k$ sparse supports and L-smooth over all $|\mathsf{S}| + 1$ sparse supports,*

$$\gamma_{\mathsf{S},k} \geq \frac{m}{L}.$$

Corollary 2. *Say the function $g(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ satisfies the assumptions of Lemma 7. Let $\mathsf{A}^\star$ be the optimal support set that maximizes $g(\cdot)$ under the sparsity constraint* (9)*. Let A_{sg} be the solution set returned by* StochasticGreedy*. Then,*

$$\mathbb{E}[f(\mathsf{A}_{sg})] \geq \left(1 - \frac{1}{e^{m/L}} - \delta\right) f(\mathsf{A}^*).$$

6.1 RSC implies Weak Subadditivity

In this section, we establish a lower bound on the subadditivity ratio in terms of only the restricted strong concavity (RSC) and smoothness (RSM) constants. This is analogous to lower bounding the submodularity ratio by Elenberg et al. [2016].

Theorem 4. *Say the function $g(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ is m strongly concave and L-smooth over all $\mathbf{x}$ supported on S. Then, the subadditivity ratio can be lower bounded as:*

$$\nu_{\mathsf{S}} \geq \frac{m}{L}. \tag{11}$$

Proof. To prove Theorem 4, we make use of the following two results. Recall that for a set S and vector $\mathbf{u}$, $\mathbf{u}_\mathsf{S}$ denotes the subvector of $\mathbf{u}$ supported on S.

Lemma 8. *For a support set* $\mathsf{S} \subset [d]$, $f(\mathsf{S}) \geq \frac{1}{2L}\|\nabla g(\mathbf{0})_\mathsf{S}\|_F^2$.

Lemma 9. *For any support set* $\mathsf{S} \subset [d]$, $f(\mathsf{S}) \leq \frac{1}{2m}\|\nabla g(\mathbf{0})_\mathsf{S}\|_F^2$.

Let A, B be a partition of a given support set $\mathsf{S} \subset [d]$ i.e $\mathsf{A} \cup \mathsf{B} = \mathsf{S}, \mathsf{A} \cap \mathsf{B} = \{\}$.

We can use Lemma 8 to lower bound the numerator of the subadditivity ratio as follows:

$$\begin{aligned} f(\mathsf{A}) + f(\mathsf{B}) &\geq \frac{1}{2L}\left(\|\nabla g(\mathbf{0})_\mathsf{A}\|_F^2 + \|\nabla g(\mathbf{0})_\mathsf{B}\|_F^2\right) \\ &= \frac{1}{2L}\|\nabla g(\mathbf{0})_\mathsf{S}\|_F^2. \end{aligned} \tag{12}$$

Combining (12) with Lemma 9, we get,

$$\nu_\mathsf{S} = \frac{f(\mathsf{A}) + f(\mathsf{B})}{f(\mathsf{S})} \geq \frac{m}{L}. \quad \square$$

Since we have a bound for the subadditivity ratio for general strongly concave and smooth functions, we can now provide a novel approximation guarantee for support selection by DISTRIBUTEDGREEDY.

Corollary 3. *Say the function* $g(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ *is* m*-strongly concave over all* $2k$ *sparse support sets and* L*-smooth over all* $k+1$ *sparse support sets. Let* $\mathsf{A}^\star$ *be the optimal support set that maximizes the sparsity constrained* $g(\cdot)$ (9). *Let* A_{dg} *be the solution set returned by* DISTRIBUTEDGREEDY. *Then,*

$$\mathbb{E}[f(\mathsf{A}_{dg})] \geq \frac{m}{2L}\left(1 - e^{-m/L}\right) f(\mathsf{A}^*).$$

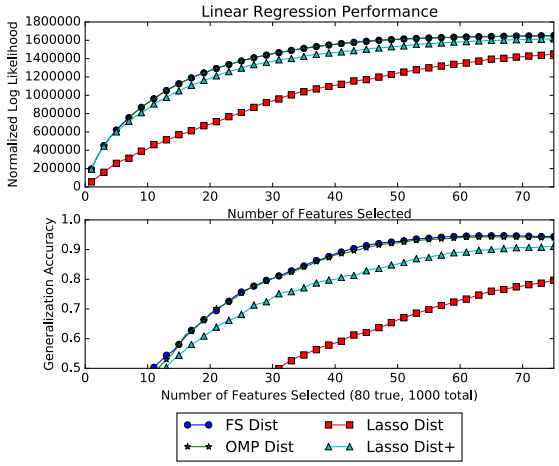
To the best of our knowledge, the bounds obtained in Corollary 3 are the first for a distributed algorithm for support selection for general functions. Note that we have taken a uniform bound for restricted strong concavity and smoothness to be over all k sized support sets, though it is only required to be over the optimal support set.

7 Experiments

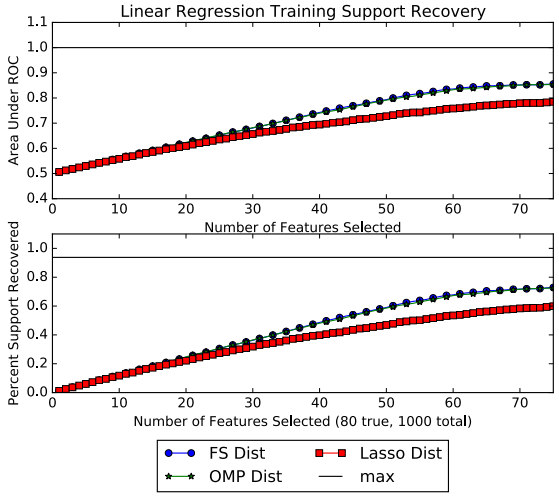
7.1 Distributed Linear Regression

We consider sparse linear regression in a distributed setting. We generate a 100-sparse regression vector is generated by selecting random nonzero entries of $\boldsymbol{\beta}$,

$$\boldsymbol{\beta}_s = (-1)^{\mathrm{Bern}(1/2)} \times \left(5\sqrt{\frac{\log d}{n}} + \delta_s\right),$$



(a)



(b)

Figure 1: Distributed linear regression, $l = 10$ partitions, $n = 800$ training and test samples, $\alpha = 0.5$. Results averaged over 10 iterations. Both greedy algorithms outperform ℓ_1 regularization.

where δ_s is a standard i.i.d. Gaussian. Measurements $\mathbf{y}$ are taken according to $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{z}$, where $\forall i \in [n], z_i$ is i.i.d. Gaussian with variance set to be $0.01\|\mathbf{X}\boldsymbol{\beta}\|_2^2$. Each row of the design matrix $\mathbf{X}$ is generated by an autoregressive process,

$$X_{n,t+1} = \sqrt{1-\alpha^2}X_{n,t} + \epsilon_{n,t},$$

where $\epsilon_{n,t}$ is i.i.d. Gaussian with variance $\alpha^2 = 0.25$. We take $n = 800$ for the number of both training and test measurements. Results are averaged over 10 iterations, each with a different partition $\{\mathsf{A}_j\}$ of the 1000 features.

We evaluate four variants of DISTRIBUTEDGREEDY. The two greedy algorithms are GREEDY Forward Selection (FS) and Orthogonal Matching Pursuit (OMP). Lasso sweeps an ℓ_1 regularization parameter λ using LARS [Efron et al.,

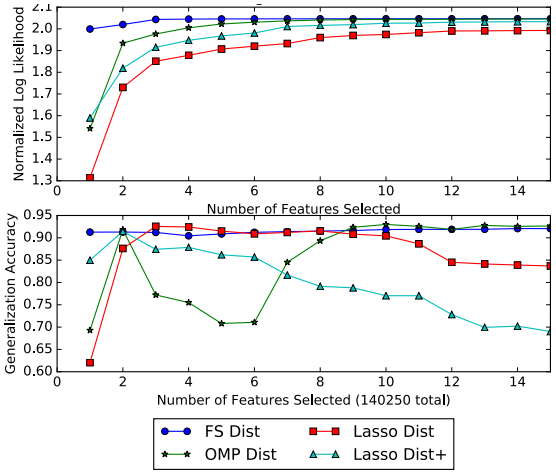


Figure 2: Distributed linear regression, Electricity dataset.

2004]. This produces nested subsets of features corresponding to a sequence of thresholds for which the support size increases by 1. Lasso uses this threshold, while Lasso+ fits an unregularized linear regression on the support set selected by Lasso.

Figure 1 shows the performance of all algorithms on the following metrics: log likelihood (normalized with respect to a null model), generalization to new test measurements from the same true support parameter, area under ROC, and percentage of the true support recovered for $l = 10$.

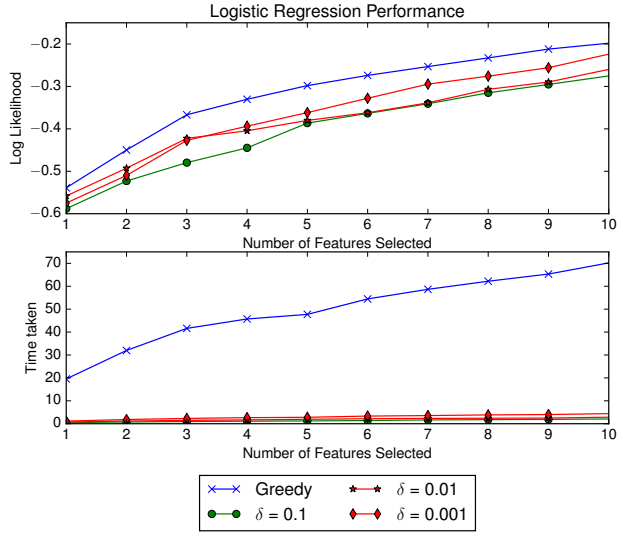


Figure 3: Trade off in time vs log likelihood for various values of δ-Stochastic Greedy as opposed to Greedy Forward Selection for logistic regression on `gisette` data [Lichman, 2013]. Results averaged over 10 iterations.

Next, we run a similar experiment on a large, real-world dataset. We sample $d = 140{,}250$ time series measurements across $n = 370$ customers from the `ElectricityLoadDiagrams` time series dataset [Lichman, 2013]. We consider the supervised learning experiment of predicting the electrical load at the *next* time 140,251. We use half of the customers for training and the rest for testing. Figure 2 shows performance of the same algorithms with data distributed across $l = 50$ partitions to select the top $k = 15$ features. We see that distributed Forward Selection produces both largest likelihood and highest generalization score. OMP has second largest likelihood, but its generalization varies widely for different values of k. This is likely due to the random placement of predictive features across a large number of partitions.

7.2 Stochastic Sparse Logistic Regression

In this section we demonstrate the applicability of Algorithm 3 for greedy support selection for sparse logistic regression. Note that the respective set function (10) when $g(\cdot)$ is the log likelihood for logistic regression is not submodular. As such one would not typically apply the StochasticGreedy algorithm for sparse logistic regression. However, the guarantees obtained in Section 6 suggest good practical performance which is indeed demonstrated in Figure 3. For the experiment we use the `gisette` dataset obtained from the UCI website [Lichman, 2013]. The dataset is of a handwritten digit recognition problem to separate out digits '4' and '9'. It has 13500 instances, and 5000 features. Figure 3 illustrates depicts the tradeoff between the time taken to learn the model and the respective training log likelihood for different values of δ as used in Algorithm 3. As shown, we obtain tremendous speed up with relatively little loss in the log likelihood value even for reasonably large δ values.

8 Conclusion

We provided novel bounds for two greedy algorithm variants for maximizing weakly submodular functions with applications to linear regression and general concave functions. Our research opens questions on what other known algorithms with provable guarantees for submodular functions can be extended to weakly submodular functions.

Acknowledgement

Research supported by NSF Grants CCF 1344179, 1344364, 1407278, 1422549, IIS 1421729, and ARO YIP W911NF-14-1-0258.

References

Rafael Barbosa, Alina Ene, Huy Le Nguyen, and Justin Ward. The power of randomization: Distributed submodular maximization on massive datasets. In *ICML*, 2015.

Aditya Bhaskara, Afshin Rostamizadeh, Jason Altschuler, Morteza Zadimoghaddam, Thomas Fu, and Vahab Mirrokni. Greedy column subset selection: New bounds and distributed algorithms. In *ICML*, 2016.

Abhimanyu Das and David Kempe. Submodular meets Spectral: Greedy Algorithms for Subset Selection, Sparse Approximation and Dictionary Selection. In *ICML*, February 2011.

Bradley Efron, Trevor Hastie, Ian Johnstone, and Robert Tibshirani. Least Angle Regression. *Annals of Statistics*, 32(2):407–499, 2004.

Ethan R. Elenberg, Rajiv Khanna, Alexandros Dimakis, and Sahand Negahban. Restricted Strong Convexity Implies Weak Submodularity. 2016.

Ahmed K. Farahat, Ahmed Elgohary, Ali Ghodsi, and Mohamed S. Kamel. Greedy column subset selection for large-scale data sets. *CoRR*, abs/1312.6838, 2013.

Anupam Gupta, Aaron Roth, Grant Schoenebeck, and Kunal Talwar. Constrained non-monotone submodular maximization: Offline and secretary algorithms. *CoRR*, abs/1003.1517, 2010.

Thibaut Horel and Yaron Singer. Maximization of approximately submodular functions. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3045–3053. Curran Associates, Inc., 2016.

Rajiv Khanna, Joydeep Ghosh, Russell A. Poldrack, and Oluwasanmi Koyejo. Sparse submodular probabilistic PCA. In *AISTATS*, 2015.

Been Kim, , and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in Neural Information Processing Systems 29*, pages 2280–2288. 2016.

Oluwasanmi Koyejo, Rajiv Khanna, Joydeep Ghosh, and Poldrack Russell. On prior distributions and approximate inference for structured variables. In *NIPS*, 2014.

Andreas Krause and Volkan Cevher. Submodular dictionary selection for sparse representation. In *ICML*, 2010.

Ravi Kumar, Benjamin Moseley, Sergei Vassilvitskii, and Andrea Vattani. Fast greedy algorithms in mapreduce and streaming. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '13, pages 1–10, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1572-2. doi: 10.1145/2486159.2486168.

Moshe Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Vahab Mirrokni and Morteza Zadimoghaddam. Randomized Composable Core-sets for Distributed Submodular Maximization. In *STOC '15*, pages 153–162, New York, New York, USA, 2015. ACM Press.

Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause 0001. Distributed Submodular Maximization - Identifying Representative Elements in Massive Data. *NIPS*, pages 2049–2057, 2013.

Baharan Mirzasoleiman, Ashwinkumar Badanidiyuru, Amin Karbasi, Jan Vondrák, and Andreas Krause. Lazier Than Lazy Greedy. *AAAI*, 2015.

Deanna Needell and Joel Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. Technical report, California Institute of Technology, Pasadena, 2008.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, December 1978.

Joel Tropp. Greed is Good: Algorithmic Results for Sparse Approximation. *IEEE Trans. Inform. Theory*, 50(10): 2231–2242, October 2004.