Journal of Machine Learning Research 19 (2018) 1–34 Submitted 10/16; Revised 8/18; Published 8/18

Approximate Submodularity and its Applications: Subset Selection, Sparse Approximation and Dictionary Selection*

Abhimanyu Das†
ABHI.DAS@GMAIL.COM
Google

David Kempe‡
DAVID.M.KEMPE@GMAIL.COM
Department of Computer Science
University of Southern California

Editor: Jeff Bilmes

Abstract

We introduce the *submodularity ratio* as a measure of how "close" to submodular a set function f is. We show that when f has submodularity ratio γ, the greedy algorithm for maximizing f provides a $(1 - e^{-\gamma})$-approximation. Furthermore, when γ is bounded away from 0, the greedy algorithm for minimum submodular cover also provides essentially an $O(\log n)$ approximation for a universe of n elements.

As a main application of this framework, we study the problem of selecting a subset of k random variables from a large set, in order to obtain the best linear prediction of another variable of interest. We analyze the performance of widely used greedy heuristics; in particular, by showing that the submodularity ratio is lower-bounded by the smallest $2k$-sparse eigenvalue of the covariance matrix, we obtain the strongest known approximation guarantees for the Forward Regression and Orthogonal Matching Pursuit algorithms.

As a second application, we analyze greedy algorithms for the dictionary selection problem, and significantly improve the previously known guarantees. Our theoretical analysis is complemented by experiments on real-world and synthetic data sets; in particular, we focus on an analysis of how tight various spectral parameters and the submodularity ratio are in terms of predicting the performance of the greedy algorithms.

1. Introduction

Over the past 10–15 years, submodularity has established itself as one of the workhorses of the Machine Learning community. A function f mapping sets to real numbers is called *submodular* if $f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$ whenever $S \subseteq T$. One of the most popular consequences of submodularity is that greedy algorithms perform quite well for maximizing the function subject to a cardinality constraint. Specifically, suppose that f is non-negative, monotone, and submodular, and consider the algorithm that, for k iterations,

*. A preliminary version was included in the proceedings of ICML 2011 under the title "Submodular Meets Spectral: Greedy Algorithms for Subset Selection, Sparse Approximation and Dictionary Selection."

†. Work done while the author was at the University of Southern California, supported in part by NSF grant DDDAS-TMRP 0540420.

‡. Supported in part by NSF CAREER award 0545855, and NSF grant DDDAS-TMRP 0540420.

adds the element x_{i+1} that has largest marginal gain $f(S_i \cup \{x_{i+1}\}) - f(S_i)$ with respect to the current set S_i. By a classic result of Nemhauser et al. (1978), this algorithm guarantees that the final set achieves a function value within a factor $1 - 1/\mathrm{e}$ of the optimum set S^* of cardinality k.

This approximation guarantee has been applied in a large number of settings; see, e.g., a survey in (Krause and Golovin, 2014). Of course, greedy algorithms are also popular when the objective function is not submodular. Typically, when f is not submodular, the greedy algorithm, though perhaps still useful in practice, will not provide theoretical performance guarantees. However, one might suspect that when f is "close to" submodular, then the performance of the greedy algorithm should degrade gracefully.

In the present article (Section 2), we formalize this intuition by defining a measure of "approximate submodularity" which we term *submodularity ratio*, and denote by γ. We prove that when a function f has submodularity ratio γ, the greedy algorithm gives a $(1 - \mathrm{e}^{-\gamma})$-approximation; in particular, whenever γ is bounded away from 0, the greedy algorithm guarantees a solution within a constant factor of optimal. We also show that for the complementary *Minimum Submodular Cover* problem, where the goal is to find the smallest set S with $f(S) \geq C$ for a given value C, the greedy algorithm gives essentially an $O(\log n)$ approximation when γ is bounded away from 0.

Subset Selection for Regression. To illustrate the usefulness of the approximate submodularity framework, we analyze greedy algorithms for the problem of *Subset Selection for Regression*: select a subset of k variables from a given set of n observation variables which, taken together, "best" predict another variable of interest. This problem has many applications ranging from feature selection, sparse learning and dictionary selection in machine learning, to sparse approximation and compressed sensing in signal processing. From a machine learning perspective, the variables are typically features or observable attributes of a phenomenon, and we wish to predict the phenomenon using only a small subset from the high-dimensional feature space. In signal processing, the variables usually correspond to a collection of dictionary vectors, and the goal is to parsimoniously represent another (output) vector. For many practitioners, the prediction model of choice is linear regression, and the goal is to obtain a linear model using a small subset of variables, to minimize the mean square prediction error or, equivalently, maximize the squared multiple correlation R^2 (Johnson and Wichern, 2002; Miller, 2002).

Thus, we formulate the Subset Selection problem for Regression as follows: Given the (normalized) covariances between n variables X_i (which can in principle be observed) and a variable Z (which is to be predicted), select a subset of $k \ll n$ of the variables X_i and a linear prediction function of Z from the selected X_i that maximizes the R^2 fit. (A formal definition is given in Section 3.) The covariances are usually obtained empirically from detailed past observations of the variable values.

The above formulation is known (see, e.g., (Das and Kempe, 2008)) to be equivalent to the problem of *sparse approximation* over dictionary vectors: the input consists of a dictionary of n feature vectors $\mathbf{x_i} \in \mathbb{R}^m$, along with a target vector $\mathbf{z} \in \mathbb{R}^m$, and the goal is to select at most k vectors whose linear combination best approximates $\mathbf{z}$. The pairwise

covariances of the previous formulation are then exactly the inner products of the dictionary vectors.[1]

This problem is **NP**-hard (Natarajan, 1995), so no polynomial-time algorithms are known to solve it optimally for all inputs. Two approaches are frequently used for approximating such problems: greedy algorithms (Miller, 2002; Tropp, 2004; Gilbert et al., 2003; Zhang, 2008) and convex relaxation schemes (Tibshirani, 1996; Candès et al., 2005; Tropp, 2006; Donoho, 2005). For our formulation, a disadvantage of convex relaxation techniques is that they do not provide explicit control over the target sparsity level k of the solution; additional effort is needed to tune the regularization parameter.

Greedy algorithms are widely used in practice for subset selection problems; for example, they are implemented in all commercial statistics packages. They iteratively add or remove variables based on simple measures of fit with Z. Two of the most well-known and widely used greedy algorithms are the subject of our analysis: Forward Regression (Miller, 2002) and Orthogonal Matching Pursuit (Tropp, 2004). (These algorithms are defined formally in Section 3).

Our main result is that using the approximate submodularity framework, approximation guarantees much stronger than all previously known bounds can be obtained quite immediately. Specifically, we show that the relevant submodularity ratio for the R^2 objective is lower-bounded by the smallest $(2k)$-sparse eigenvalue $\lambda_{\min}(C, 2k)$ of the covariance matrix C of the observation variables. Combined with our general bounds for approximately submodular functions, this immediately implies a $(1 - \mathrm{e}^{-\lambda_{\min}(C,2k)})$-approximation guarantee for Forward Regression. For Orthogonal Matching Pursuit, a similar analysis leads to a somewhat weaker guarantee of essentially $(1 - \mathrm{e}^{-\lambda_{\min}(C,2k)^2})$. In a precise sense, our analysis thus shows that the less singular C (or its small principal submatrices) are, the "closer to" submodular the R^2 objective. Previously, Das and Kempe (2008) had shown that R^2 is truly submodular when there are no "conditional suppressor" variables; however, the latter is a much stronger condition.

Most previous results for greedy subset selection algorithms (e.g., (Gilbert et al., 2003; Tropp, 2004; Das and Kempe, 2008)) had been based on coherence of the input data, i.e., the maximum correlation μ between any pair of variables. Small coherence is an extremely strong condition, and the bounds usually break down when the coherence is $\omega(1/k)$. On the other hand, most bounds for greedy and convex relaxation algorithms for sparse recovery are based on a weaker sparse-eigenvalue or Restricted Isometry Property (RIP) condition (Zhang, 2009, 2008; Lozano et al., 2009; Zhou, 2009; Candès et al., 2005). However, these results apply to a different objective: minimizing the difference between the actual and estimated coefficients of a sparse vector. Simply extending these results to the subset selection problem adds a dependence on the largest k-sparse eigenvalue and only leads to weak additive bounds.

Dictionary Selection. As a second illustration of the approximate submodularity framework, we obtain much tighter theoretical performance guarantees for greedy algorithms for dictionary selection (Krause and Cevher, 2010). In the *Dictionary Selection problem*, we are given s target vectors, and a candidate set V of feature vectors. The goal is to select a set

1. For this reason, the dimension m of the feature vectors only affects the problem indirectly, via the accuracy of the estimated covariance matrix.

$D \subset V$ of at most d feature vectors, which will serve as a *dictionary* in the following sense. For each of the target vectors, the best $k < d$ vectors from D will be selected and used to achieve a good R^2 fit; the goal is to maximize the average R^2 fit for all of these vectors. (A formal definition is given in Section 4.) The problem of finding a dictionary of basis functions for sparse representation of signals has several applications in machine learning and signal processing. Krause and Cevher (2010) showed that greedy algorithms for dictionary selection can perform well in many instances, and proved additive approximation bounds for two specific algorithms, SDS_{MA} and SDS_{OMP} (defined in Section 4). Our approximate submodularity framework directly yields stronger multiplicative approximation guarantees.

Our theoretical analysis is complemented by experiments comparing the performance of the greedy algorithms and a baseline convex-relaxation algorithm for subset selection on two real-world data sets and a synthetic data set. We also evaluate the submodularity ratio of these data sets and compare it with other spectral parameters: while the input covariance matrices are close to singular, the submodularity ratio actually turns out to be significantly larger.

While the submodularity ratio is always *lower-bounded* by the smallest (sparse) eigenvalue, our experiments reveal that this lower bound can be loose. This happens when there are small (sparse) eigenvalues, but the predictor variable is not badly aligned with their eigenspace. Hence, computing the submodularity ratio explicitly (although it appears computationally intensive to do so) can lead to stronger post hoc approximation guarantees. In this context, we also discuss ways in which a more careful analysis of the greedy algorithms allows significantly stronger post hoc approximation guarantees.

Our main contributions can be summarized as follows:

1. We introduce (in Section 2) the notion of the submodularity ratio as a predictor of the performance of greedy algorithms. We show that a submodularity ratio of γ leads to a $(1 - \mathrm{e}^{-\gamma})$-approximation guarantee for the greedy algorithm for maximum coverage. For the minimum cover probem, we show essentially a $\frac{\log n}{\gamma}$ approximation guarantee for the greedy algorithm.

2. Using the approximate submodularity framework, in Section 3, we obtain the strongest known theoretical performance guarantees for greedy algorithms for subset selection. In particular, we show that the Forward Regression and OMP algorithms are within a $1 - \mathrm{e}^{-\gamma}$ factor and $1 - \mathrm{e}^{-(\gamma \cdot \lambda_{\min})}$ factor of the optimal solution, respectively (where the γ and λ terms are appropriate submodularity and sparse-eigenvalue parameters).

3. Again using the approximate submodularity framework, in Section 4, we obtain the strongest known theoretical guarantees for algorithms for dictionary selection, improving on the results of Krause and Cevher (2010). In particular, we show that the SDS_{MA} algorithm is within a factor $\frac{\gamma}{\lambda_{\max}}(1 - \frac{1}{\mathrm{e}})$ of optimal.

4. We evaluate our theoretical bounds for subset selection by running greedy and L1-relaxation algorithms on real-world and synthetic data, and show how the various submodular and spectral parameters correlate with the performance of the algorithms in practice.

1.1 Related and Subsequent Work

We provide an overview of related work both in the context of subset selection (and its variants) and in submodular optimization, as well as a discussion of work that appeared subsequent to the conference version of the present article.

1.1.1 Subset Selection and Sparse Recovery

There has been a lot of related work in the statistics, machine learning and signal processing communities on problems with sparsity constraints (such as sparse recovery, compressed sensing, sparse approximation and feature selection).

In sparse recovery, one is given an $n \times m$ dictionary ϕ of m vectors in $\mathbb{R}^n$ (where $n < m$), along with another vector $y \in \mathbb{R}^n$. It is known that y has some sparse representation in terms of k vectors of ϕ, up to a small noise term ϵ, and the goal is to recover the coefficients α given y, ϕ and ϵ. There has been a lot of recent interest in greedy and convex relaxation techniques for the sparse recovery problems, both in the noiseless and noisy setting. For L1 relaxation techniques, Tropp (2006) showed conditions based on the coherence (i.e., the maximum correlation between any pair of variables) of the dictionary that guaranteed near-optimal recovery of a sparse signal. In (Candès et al., 2005; Donoho, 2005), it was shown that if the target signal is truly sparse, and the dictionary obeys a Restricted Isometry Property (RIP), then L1 relaxation can almost exactly recover the true sparse signal. Other results (Zhao and Yu, 2006; Zhou, 2009) also prove conditions under which L1 relaxation can recover a sparse signal. Though related, the above results are not directly applicable to our subset selection formulation, since the goal in sparse recovery is to recover the true coefficients of the sparse signal, as opposed to our problem of minimizing the prediction error of an arbitrary signal subject to a specified sparsity level.

For greedy sparse recovery, Zhang (2008, 2009) and Lozano et al. (2009) provided conditions based on sparse eigenvalues under which Forward Regression and Forward-Backward Regression can recover a sparse signal. As with the L1 results for sparse recovery, the objective function analyzed in these papers is somewhat different from that in our subset selection formulation; furthermore, these results are intended mainly for the case when the predictor variable is truly sparse. Simply extending these results to our problem formulation gives weaker, additive bounds and requires stronger conditions than our results.

The papers by Das and Kempe (2008), Gilbert et al. (2003) and Tropp et al. (2003); Tropp (2004) analyzed greedy algorithms for sparse approximation, which as mentioned previously is equivalent to our subset selection formulation presented in this work. In particular, they obtained a $1 + \Theta(\mu^2 k)$ multiplicative approximation guarantee for the mean square error objective and a $1 - \Theta(\mu k)$ guarantee for the R^2 objective, whenever the coherence μ of the dictionary is $O(1/k)$. These results are thus weaker than those presented here, since they do not apply to instances with even moderate correlations of $\omega(1/k)$.

Other analysis of greedy methods includes the work of Natarajan (1995), which proved a bicriteria approximation bound for minimizing the number of vectors needed to achieve a given prediction error.

As mentioned earlier, the paper by Krause and Cevher (2010) analyzed greedy algorithms for the dictionary selection problem, which generalizes subset selection to prediction of multiple variables. They too use a notion of approximate submodularity to provide ad-

ditive approximation guarantees. Since their analysis is for a more general problem than subset selection, applying their results directly to the subset selection problem predictably gives much weaker bounds than those presented in this paper for subset selection. Furthermore, even for the general dictionary selection problem, our techniques can be used to significantly improve their analysis of greedy algorithms and obtain tighter multiplicative approximation bounds (as shown in Section 4).

In general, we note that the performance bounds for greedy algorithms derived using the coherence parameter are usually the weakest, followed by those using the Restricted Isometry Property, then those using sparse eigenvalues, and finally those using the submodularity ratio. (We show an empirical comparison of these parameters in Section 5.)

1.1.2 Submodular Maximization and Curvature

In the context of submodular maximization, the celebrated result of Nemhauser et al. (1978) proved that the greedy algorithm obtained a $(1 - 1/e)$-approximation for maximizing any monotone, submodular set function subject to a uniform matroid. The same guarantee was obtained by Calinescu et al. (2011) for an arbitrary matroid constraint, using a continuous variant of the greedy algorithm.

While we are not aware of prior work on defining a notion of how far a function is from being submodular (or analyzing greedy algorithms for such functions), there is a well-known notion of curvature (Conforti and Cornuéjols, 1984; Vondrák, 2010) that captures how far a submodular function is from being *modular*. In particular, the *total curvature* of a submodular set function is defined as $c = 1 - \min_{S, j \notin S} \frac{f_S(j)}{f_\emptyset(j)}$, where $f_S(j) = f(S \cup j) - f(S)$. (Additional related notions include average curvature and monotonicity ratio; see (Iyer, 2015) for a discussion.) Intuitively c measures how far away f is from being modular, and is equal to 0 if f is modular. Conforti and Cornuéjols (1984) analyzed the greedy algorithm for submodular maximization in terms of the c parameter, and showed a $\frac{1}{c}(1-e^{-c})$ approximation for a uniform matroid. The result was extended to an arbitrary matroid by Vondrák (2010), and an improved guarantee of $(1-c/e)$ was obtained recently by Sviridenko et al. (2015). Curvature was also used by Iyer et al. (2013) to obtain improved bounds for submodular function approximation, PMAC-learning and submodular minimization.

Another notion of approximate modularity was recently proposed by Chierichetti et al. (2015), who defined a function to be ϵ-approximately modular if it satisfies all the modularity requirements to within an ϵ additive error. Chierichetti et al. (2015) analyzed how close (in the l_∞ distance) any approximately modular function can be to a modular function.

Note that both the notions of total curvature and approximate modularity are different from the submodularity ratio proposed in this paper, which measures how far a set function is from being submodular.

1.1.3 Subsequent Work

Subsequent to our work introducing the submodularity ratio, several papers have used this notion for analyzing greedy algorithms for machine learning applications. Das et al. (2012) proposed diversity-promoting spectral regularizers for feature selection, and used the submodularity ratio to analyze a hybrid greedy and local search algorithm for the diverse feature selection problem. Grubb and Bagnell (2012) analyzed greedy algorithms for learning an

ensemble of *anytime predictors* that automatically trade computation time with predictive accuracy. Using the submodularity ratio, the authors provide an approximation guarantee for the performance of their ensemble algorithm. Kusner et al. (2014) analyzed greedy methods for training a tree of classifiers for feature-cost sensitive learning, and show that the objective function for obtaining a cost-sensitive tree of classifiers is approximately submodular. Qian et al. (2015) proposed a Pareto optimization approach for subset selection in sparse regression and analyzed the performance of their algorithm using the submodularity ratio.

Most directly following up on our initial work is a recent result of Elenberg et al. (2018) that extends our analysis of greedy algorithms for subset selection from the linear regression setting to arbitrary Generalized Linear Models. The main result is a lower bound on any function's submodularity ratio in terms of its restricted strong convexity and smoothness parameters, which can then be used to obtain approximation guarantees for greedy feature selection algorithms.

2. Approximate Submodularity

We begin by defining our notion of approximate submodularity, and explaining its relationship with the traditional notion of submodularity. Then, we show that approximation results for greedy algorithms degrade gracefully as the function becomes less and less submodular.

2.1 Submodularity Ratio

We introduce the notion of submodularity ratio for a general set function, which captures "how close" to submodular the function is. Let $\mathcal{X}$ be a universe of elements, and Let $f : 2^{\mathcal{X}} \to \mathbb{R}^+$ be a non-negative set function.

Definition 1 (Monotonicity, Submodularity) *1. f is* monotone *iff $f(S) \leq f(T)$ whenever $S \subseteq T$.*

2. f is submodular *iff $f(S \cup \{x\}) - f(S) \geq f(T \cup \{x\}) - f(T)$ whenever $S \subseteq T$.*

Our definition of the submodularity ratio smoothly interpolates between functions that are submodular and those that are far from so.

Definition 2 (Submodularity Ratio) *The* submodularity ratio *of a monotone function f with respect to a set U and a parameter $k \geq 1$ is*

$$\gamma_{U,k}(f) = \min_{L \subseteq U, S : |S| \leq k, S \cap L = \emptyset} \frac{\sum_{x \in S} f(L \cup \{x\}) - f(L)}{f(L \cup S) - f(L)}, \tag{1}$$

where we define $0/0 := 1$. Thus, the submodularity ratio captures how much more f can increase by adding any subset S of size k to L, compared to the combined benefits of adding its individual elements to L. That Definition 2 generalizes submodularity is captured by the following proposition.

Proposition 3 *f is submodular if and only if $\gamma_{U,k} \geq 1$ for all U and k.*

Proof. First, assume that $\gamma_{U,k} \geq 1$ for all U and k. By choosing $k = 2$ and $S = \{x, y\}$ in Equation (1), we obtain that $f(L \cup \{x\}) + f(L \cup \{y\}) \geq f(L \cup \{x, y\}) + f(L)$, or, rearranged, $f(L \cup \{x\}) - f(L) \geq f(L \cup \{x, y\}) - f(L \cup \{y\})$. Now, when we have two sets S and $T = S \cup \{x_1, x_2, \ldots, x_k\}$, define $S_i := S \cup \{x_1, \ldots, x_i\}$ for $0 \leq i \leq k$. Setting $L = S_i$ now gives us that $f(S_i \cup \{x\}) - f(S_i) \geq f(S_{i+1} \cup \{x\}) - f(S_{i+1})$. Induction on i now completes the proof.

Conversely, assume that f is submodular. In Equation (1), let $S = \{x_1, \ldots, x_k\}$ and $S_i = \{x_1, \ldots, x_i\}$, and write a telescoping series $f(L \cup S) - f(L) = \sum_{i=0}^{k-1} f(L \cup S_{i+1}) - f(L \cup S_i)$. By submodularity of f, we can bound

$$f(L \cup S_{i+1}) - f(L \cup S_i) \;=\; f(L \cup S_i \cup \{x_{i+1}\}) - f(L \cup S_i) \;\leq\; f(L \cup \{x_{i+1}\}) - f(L),$$

which gives us a lower bound of 1 on the ratio. ■

Remark 4 *The submodularity ratio is defined as a minimum over exponentially many values, and in general, it is* **NP***-hard to compute exactly (more recently, Bai and Bilmes (2018) showed that it cannot be computed in polynomial time in the value oracle model). This is a property it shares with the well-known Restricted Isometry Property (RIP) (Candès and Tao, 2005): computing the RIP of a matrix is essentially equivalent to computing the expansion of a graph, yet the guarantees for sparse approximation algorithms are frequently expressed in terms of the RIP.*

Whether one can efficiently approximate the submodularity ratio to within non-trivial factors is an interesting open question. Approximating it would allow one to at least derive post hoc approximation guarantees, i.e., to give the user guarantees on the approximation quality for the specific instance that was solved. In the appendix, we discuss some (fairly strong) assumptions under which one can derive non-trivial lower bounds on the submodularity ratio.

Typically, rather than computing the submodularity ratio on a given instance, one would use problem-specific insights to derive a priori lower bounds on the submodularity ratio in terms of quantities that are easier to compute exactly or approximately. For example, in the primary application studied here (linear regression), the submodularity ratio is lower-bounded by the (easy to compute) smallest eigenvalue of the covariance matrix, and more tightly bounded by the (not so easy to compute) smallest $2k$-sparse eigenvalue of the covariance matrix. Recently, Elenberg et al. (2018) showed how to derive similar lower bounds for a more general class of linear objective functions. We anticipate that similar types of bounds can be obtained for other classes of objectives.

2.2 The Greedy Algorithm for Maximum Coverage

Probably the most widely used fact about (monotone) submodular functions is that a simple greedy algorithm approximately optimizes the function subject to a cardinality constraint.[2] This is a celebrated result by Nemhauser et al. (1978). Specifically, Nemhauser et al. (1978) analyzed the following algorithm.

Let S^{NG} be the final set S_k returned by the algorithm. The following theorem of Nemhauser et al. (1978) is widely used in the Machine Learning and related communities:

2. Many other algorithmic optimization problems are easier for submodular function. Some of them are discussed in Section 6.

Algorithm 1 The Nemhauser Greedy Algorithm for a non-negative, monotone, and submodular set function f on a universe $\mathcal{X}$.

1: Initialize $S_0 = \emptyset$.
2: **for** each iteration $i + 1 = 1, 2, \ldots$ **do**
3: Let $x_{i+1} \in \mathcal{X}$ be an element maximizing $f(S_i \cup \{x_{i+1}\})$, and set $S_{i+1} = S_i \cup \{x_{i+1}\}$.
4: Output S_k.

Theorem 5 (Nemhauser et al. (1978)) *The set S^{NG} returned by the Nemhauser Greedy Algorithm guarantees that $f(S^{NG}) \geq (1 - \frac{1}{\mathrm{e}}) \cdot f(S_k^*)$, where S_k^* is the set maximizing $f(S)$ among all size-k sets S.*

The centerpiece of our algorithmic analysis is a generalization of Theorem 5 to approximately submodular functions.

Theorem 6 *Let f be a nonnegative, monotone set function, and OPT be the maximum value of f obtained by any set of size k. Then, the set S^{NG} selected by the Nemhauser Greedy Algorithm has the following approximation guarantee:*

$$f(S^{NG}) \geq \left(1 - \mathrm{e}^{-\gamma_{S^{NG},k}(f)}\right) \cdot OPT.$$

Notice that for submodular functions, because $\gamma_{S^{\mathrm{NG}},k}(f) \geq 1$, our theorem recovers the result of Nemhauser et al. (1978) as a special case.

Proof. We carry out the analysis in somewhat more generality than needed here, since most of it will be useful in Section 2.3. Let k be the number of iterations that Algorithm 1 was run, and S_i^{NG} the set of elements greedily chosen in the first i iterations. Let S_i^{NG} be the set of variables chosen by the Nemhauser Greedy Algorithm (Algorithm 1) in the first i iterations. Define $A(i) = f(S_i^{\mathrm{NG}}) - f(S_{i-1}^{\mathrm{NG}})$ to be the gain obtained from the variable chosen by the algorithm in iteration i. Then $f(S^{\mathrm{NG}}) = \sum_{i=1}^{k} A(j)$.

For simplicity of notation, we write $f(x/S)$ to denote $f(\{x\} \cup S) - f(S)$, and $f(T/S)$ to denote $f(T \cup S) - f(S)$, for any element $x \in \mathcal{X}$ and sets S and T. We will also write $\gamma_{S^{\mathrm{NG}},k}$ to denote $\gamma_{S^{\mathrm{NG}},k}(f)$.

Let S^* be some (optimum) set of k^* variables, achieving a value of (at least) C. Let $S_i = S^* \setminus S_i^{\mathrm{NG}}$. By monotonicity of f and the fact that $S_i \cup S_i^{\mathrm{NG}} \supseteq S^*$, we have that $f(S_i \cup S_i^{\mathrm{NG}}) \geq C$. We will show that at least one of the $x \in S_i$ is a good candidate in iteration $i + 1$ of the algorithm. First, the joint contribution of S_i, conditioned on the set S_i^{NG}, must be fairly large: $f(S_i/S_i^{\mathrm{NG}}) = f(S_i \cup S_i^{\mathrm{NG}}) - f(S_i^{\mathrm{NG}}) \geq C - f(S_i^{\mathrm{NG}})$. Using Definition 2, as well as $S_i^{\mathrm{NG}} \subseteq S^{\mathrm{NG}}$ and $|S_i| \leq k^*$,

$$\sum_{x \in S_i} f(x/S_i^{\mathrm{NG}}) \geq \gamma_{S_i^{\mathrm{NG}},|S_i|} \cdot f(S_i/S_i^{\mathrm{NG}}) \;\geq\; \gamma_{S^{\mathrm{NG}},k^*} \cdot f(S_i/S_i^{\mathrm{NG}}).$$

Let $\hat{x} \in \mathrm{argmax}_{x \in S_i} f(x/S_i^{\mathrm{NG}})$ maximize $f(\hat{x}/S_i^{\mathrm{NG}})$. Then we get that

$$f(\hat{x}/S_i^{\mathrm{NG}}) \geq \frac{\gamma_{S^{\mathrm{NG}},k^*}}{|S_i|} \cdot f(S_i/S_i^{\mathrm{NG}}) \;\geq\; \frac{\gamma_{S^{\mathrm{NG}},k^*}}{k^*} \cdot f(S_i/S_i^{\mathrm{NG}}).$$

Since the $\hat{x}$ above was a candidate to be chosen in iteration $i+1$, and the algorithm chose a variable x_{i+1} such that $f(x_{i+1}/S_i^{\mathrm{NG}}) \geq f(x/S_i^{\mathrm{NG}})$ for all $x \notin S_i^{\mathrm{NG}}$, we obtain that

$$A(i+1) \geq \frac{\gamma_{S^{\mathrm{NG}},k^*}}{k^*} \cdot f(S_i/S_i^{\mathrm{NG}}) \;\geq\; \frac{\gamma_{S^{\mathrm{NG}},k^*}}{k^*} \cdot (C - f(S_i^{\mathrm{NG}})) \;\geq\; \frac{\gamma_{S^{\mathrm{NG}},k^*}}{k^*} \cdot (C - \sum_{j=1}^{i} A(j)).$$

We will use the above inequality to prove by induction on t that

$$C - \sum_{i=1}^{t} A(i) \leq C \cdot (1 - \frac{\gamma_{S^{\mathrm{NG}},k^*}}{k^*})^t \;\leq\; C \cdot \mathrm{e}^{-\gamma_{S^{\mathrm{NG}},k^*} \cdot \frac{t}{k^*}}. \tag{2}$$

The base case is clearly true for $t = 0$. Suppose that the inequality is true after t iterations. Then, at iteration $t+1$, we have

$$\begin{aligned}
C - \sum_{i=1}^{t+1} A(i) &= C - \sum_{i=1}^{t} A(i) - A(t+1) \\
&\leq C - \sum_{i=1}^{t} A(i) - \frac{\gamma_{S^{\mathrm{NG}},k^*}}{k^*} \cdot (C - \sum_{i=1}^{t} A(i)) \\
&= (C - \sum_{i=1}^{t+1} A(i)) \cdot \left(1 - \frac{\gamma_{S^{\mathrm{NG}},k^*}}{k^*}\right) \\
&\leq C \cdot \left(1 - \frac{\gamma_{S^{\mathrm{NG}},k^*}}{k^*}\right)^{t+1},
\end{aligned}$$

thus completing the inductive proof. Using Inequality(2) with $k = k^*, t = k-1$ and $C = \mathrm{OPT}$, we obtain that

$$f(S^{\mathrm{NG}}) = \sum_{i=1}^{k} A(i) \;\geq\; \mathrm{OPT} \cdot \left(1 - \mathrm{e}^{-\gamma_{S^{\mathrm{NG}},k}}\right).$$

This completes the proof of the approximation guarantee. ■

Remark 7 *As the submodularity ratio goes to 0, the approximation guarantee of Theorem 6 deteriorates and becomes 0 in the limit. This is not surprising: in the limit, the definition does not place any restrictions on the function f. Without any restrictions on f, not only can the greedy algorithm perform arbitrarily poorly, but the same may be true for* any *efficient algorithm, since f might be a function that is provably hard to approximate to within any non-trivial factor.*

Indeed, the goal of Theorem 6 is not to provide a universal approximation guarantee, but rather to outline conditions under which running the greedy algorithm comes with provable approximation guarantees. Practitioners run greedy algorithms routinely without any guarantees, and the submodularity ratio may provide guidance under what conditions doing so has theoretical justification, even when the objective function f is not submodular.

2.3 The Greedy Algorithm for Minimum Submodular Cover

The "complementary" problem to submodular function maximization is minimum submodular cover, where the goal is to find a smallest set S with $f(S) \geq C$, a given target value. The name derives from one of the most common instance of submodular functions: coverage functions.[3] Here, the elements x correspond to sets, and the function value f is the size of the union of the selected sets. In the Maximum Coverage Problem, the goal is to maximize the size of the union by selecting k sets, and in the Minimum Set Cover Problem, the goal is to cover all elements selecting as few sets as possible.

For both problems, the greedy algorithm (Algorithm 1) provides essentially best possible guarantees. The only difference is the termination condition: for maximum coverage, the algorithm is terminated when k sets are selected, while for minimum cover, the algorithm is terminated when all elements (or a given number) have been covered. For the Minimum Set Cover Problem, the greedy algorithm achieves a $\ln n$ approximation, which is best possible unless $\mathbf{P} = \mathbf{NP}$. For more general monotone submodular functions, the results are somewhat less clean to express, but are summarized by the following theorem of Wolsey (1982).

Theorem 8 (Theorem 1 of Wolsey (1982)) *Let f be nonnegative, monotone and submodular, and let $n = |\mathcal{X}|$. For any given C, let $k^*(C)$ be the size of the smallest set $S \subseteq V$ such that $f(S) \geq C$. Let k be the size of the set S^{NG} selected by Algorithm 1 when run until $f(S) \geq C$. Then,*

$$k \leq \left(1 + \log\left(\frac{C}{C - f(S^{NG}_{k-1})}\right)\right) \cdot k^*(C),$$

where S^{NG}_{k-1} is the set selected by Algorithm 1 after $k-1$ iterations.

If f is integer valued, then

$$k \leq (1 + \log(\theta)) \cdot k^*,$$

where $\theta = \max_{x \in \mathcal{X}} f(x)$ is the maximum value of the set function obtained by a single element.

We show that Theorem 8, too, extends gracefully to approximately submodular functions f.

Theorem 9 *Let f be a nonnegative and monotone function, and let $n = |\mathcal{X}|$. For any given C, let $k^*(C)$ be the size of the smallest set $S \subseteq V$ such that $f(S) \geq C$. Let k be the size of the set S^{NG} selected by Algorithm 1 when run until $f(S) \geq C$. Then,*

$$k \leq 1 + \frac{1}{\gamma_{S^{NG},k^*(C)}(f)} \cdot \log\left(\frac{C}{C - f(S^{NG}_{k-1})}\right) \cdot k^*(C),$$

where S^{NG}_{k-1} is the set selected by Algorithm 1 after $k-1$ iterations.

3. A characterization of coverage functions in terms of functional properties akin to submodularity is given by Salek et al. (2010).

Proof. We use the same notation as in the proof of Theorem 6. For notational convenience, write $k^* = k^*(C)$. Let k be the number of iterations taken by Algorithm 1, so that $f(S^{\mathrm{NG}}_k) \geq C$ and $f(S^{\mathrm{NG}}_{k-1}) < C$. Thus $f(S^{\mathrm{NG}}) = \sum_{j=1}^{k} A(j)$.

Let S^* be a smallest set (i.e., $|S^*| = k^*$) with $f(S^*) \geq C$. Substituting $t = k-1$ into Equation (2) and solving for k, we obtain that

$$k \leq 1 + \frac{1}{\gamma_{S^{\mathrm{NG}},k^*}(f)} \cdot \log\left(\frac{C}{C - f(S^{\mathrm{NG}}_{k-1})}\right) \cdot k^*,$$

as claimed. ■

As with Wolsey's result for submodular functions, the bounds can be improved when f is integer-valued.

Theorem 10 *Assume that f is integer-valued, in addition to all conditions (and notation) of Theorem 9. Let $\theta = \max_{x \in \mathcal{X}} f(x)$ is the maximum value of the set function obtained by any single element. Then, the number k of elements selected by Algorithm 1 satisfies*

$$k \leq 1 + \frac{1}{\gamma_{S^{NG},k^*(C)}(f)} \cdot \log(C) \cdot k^*(C),$$
$$k \leq \left(1 + \frac{1}{\gamma_{S^{NG},k^*(C)}(f)} \log\left(\frac{\theta}{\gamma_{\emptyset,k^*}(f)}\right)\right) \cdot k^*(C).$$

Proof. The first result follow directly from Theorem 9, because $C - f(S^{\mathrm{NG}}_{k-1}) \geq 1$ for integer-valued functions.

For the second result, substitute $t = \frac{k^*}{\gamma_{S^{\mathrm{NG}},k^*}(f)} \cdot \log\left(\frac{f(S^*)}{k^*}\right)$ into Inequality (2) to obtain that

$$C - f(S^{\mathrm{NG}}_t) \leq C \cdot \mathrm{e}^{-\frac{\gamma_{S^{\mathrm{NG}},k^*}}{k^*} \cdot t} \;\leq\; k^*.$$

Because f is a monotone and integer-valued, $f(S^{\mathrm{NG}}_i) - f(S^{\mathrm{NG}}_{i-1}) \geq 1$ for all remaining iterations i, and it takes at most k^* additional iterations to reach a value of C. Hence,

$$k \leq t + k^* \;=\; \left(1 + \frac{1}{\gamma_{S^{\mathrm{NG}},k^*}(f)} \cdot \log(C/k^*)\right) \cdot k^* \;\leq\; \left(1 + \frac{1}{\gamma_{S^{\mathrm{NG}},k^*}(f)} \cdot \log\left(\frac{\theta}{\gamma_{\emptyset,k^*}(f)}\right)\right) \cdot k^*.$$

The inequality $C/k^* \leq \theta/\gamma_{\emptyset,k^*}(f)$ is directly from Definition 2. ■

The same techniques can be used to obtain the following bicriteria approximation guarantee below. The bicriteria guarantees are similar in spirit to, for instance, (Krause and Golovin, 2014, Theorem 1.5). We believe that similar results for submodular functions are folklore among researchers, though we are unaware of a reference stating precisely the form we give here.

Theorem 11 *For any $\epsilon \in (0,1)$, if Algorithm 1 is run until $f(S^{NG}) \geq (1-\epsilon) \cdot C$, the size of the set S^{NG} that is returned is at most $\frac{1}{\gamma_{S^{NG},k^*}(f)} \log(\frac{1}{\epsilon}) \cdot k^*(C)$.*

Proof. For the proof, simply substitute $t = \frac{1}{\gamma_{S^{\mathrm{NG}},k^*}(f)} \log(\frac{1}{\epsilon}) \cdot k^*(C)$ into Inequality (2). ■

A particularly clean corollary of this theorem is obtained when $\epsilon = 1/\mathrm{e}$. In that case, we obtain a $(1 - 1/\mathrm{e})$ approximation by increasing the set size by a factor $\frac{1}{\gamma_{S^{\mathrm{NG}},k^*}(f)}$. Thus, instead of a smooth degradation of the customary $(1 - 1/\mathrm{e})$ approximation guarantee, we can choose a smooth increase in the size of the set that the greedy algorithm is allowed to select, and thus retain the customary $(1 - 1/\mathrm{e})$ approximation, even for functions that are only approximately submodular.

3. Subset Selection for Regression

As our first and main application of the approximate submodularity framework, we analyze greedy algorithms for subset selection in regression. The goal in subset selection is to estimate a *predictor variable* Z using linear regression on a small subset from the set of *observation variables* $\mathcal{X} = \{X_1, \ldots, X_n\}$. We use $\mathrm{Var}[X_i]$, $\mathrm{Cov}[X_i, X_j]$ and $\rho(X_i, X_j)$ to denote the variance, covariance and correlation of random variables, respectively. By appropriate normalization, we can assume that all the random variables have expectation 0 and variance 1. The matrix of covariances between the X_i and X_j is denoted by C, with entries $c_{i,j} = \mathrm{Cov}[X_i, X_j]$. Similarly, we use $\mathbf{b}$ to denote the covariances between Z and the X_i, with entries $b_i = \mathrm{Cov}[Z, X_i]$. Formally, the *Subset Selection* problem can now be stated as follows:

Definition 12 (Subset Selection) *Given pairwise covariances among all variables, as well as a parameter k, find a set $S \subset \mathcal{X}$ of at most k variables X_i and a linear predictor $Z' = \sum_{i \in S} \alpha_i X_i$ of Z, maximizing the* squared multiple correlation *(Diekhoff, 2002; Johnson and Wichern, 2002)*

$$R^2_{Z,S} = \frac{\mathrm{Var}[Z] - \mathbb{E}\left[(Z - Z')^2\right]}{\mathrm{Var}[Z]}.$$

R^2 is a widely used measure for the goodness of a statistical fit; it captures the fraction of the variance of Z explained by variables in S. Because we assumed Z to be normalized to have variance 1, it simplifies to $R^2_{Z,S} = 1 - \mathbb{E}\left[(Z - Z')^2\right]$.

For a set S, we use C_S to denote the submatrix of C with row and column set S, and $\mathbf{b}_S$ to denote the vector with only entries b_i for $i \in S$. For notational convenience, we frequently do not distinguish between the index set S and the variables $\{X_i \mid i \in S\}$. Given the subset S of variables used for prediction, the optimal regression coefficients α_i are well known to be $\mathbf{a}_S = (\alpha_i)_{i \in S} = C_S^{-1} \cdot \mathbf{b}_S$ (see, e.g., (Johnson and Wichern, 2002)), and hence $R^2_{Z,S} = \mathbf{b}_S^{\intercal} C_S^{-1} \mathbf{b}_S$. Thus, the subset selection problem can be phrased as follows: Given C, $\mathbf{b}$, and k, select a set S of at most k variables to maximize $R^2_{Z,S} = \mathbf{b}_S^{\intercal}(C_S^{-1})\mathbf{b}_S$.[4]

Many of our results are phrased in terms of eigenvalues of the covariance matrix C and its submatrices. Covariance matrices are positive semidefinite, so their eigenvalues are real

4. We assume throughout that C_S is non-singular. For some of our results, an extension to singular matrices is possible using the Moore-Penrose generalized inverse.

and non-negative (Johnson and Wichern, 2002). We denote the eigenvalues of a positive semidefinite matrix A by $\lambda_{\min}(A) = \lambda_1(A) \leq \lambda_2(A) \leq \cdots \leq \lambda_n(A) = \lambda_{\max}(A)$. We use $\lambda_{\min}(C,k) = \min_{S:|S|=k} \lambda_{\min}(C_S)$ to refer to the smallest eigenvalue of any $k \times k$ submatrix of C (i.e., the smallest k-sparse eigenvalue), and similarly $\lambda_{\max}(C,k) = \max_{S:|S|=k} \lambda_{\max}(C_S)$.[5] We also use $\kappa(C,k)$ to denote the largest condition number (the ratio of the largest and smallest eigenvalue) of any $k \times k$ submatrix of C. This quantity is strongly related to the Restricted Isometry Property in (Candès et al., 2005). We also use $\mu(C) = \max_{i \neq j} |c_{i,j}|$ to denote the *coherence*, i.e., the maximum absolute pairwise correlation between the X_i variables. Recall the L_2 vector and matrix norms: $\|\mathbf{x}\|_2 = \sqrt{\sum_i |x_i|^2}$, and $\|A\|_2 = \lambda_{\max}(A) = \max_{\|\mathbf{x}\|_2=1} \|A\mathbf{x}\|_2$. We also use $\|\mathbf{x}\|_0 = |\{i \mid x_i \neq 0\}|$ to denote the sparsity of a vector $\mathbf{x}$.

The Rayleigh-Ritz representation for $\|A\|_2$ is useful in bounding $\lambda_{\min}(A)$, as for any vector $\mathbf{x}$, we have $\lambda_{\min}(A) \leq \frac{\|A\mathbf{x}\|_2}{\|\mathbf{x}\|_2}$.

The part of a variable Z that is not correlated with the X_i for all $i \in S$, i.e., the part that cannot be explained by the X_i, is called the *residual* (see (Diekhoff, 2002)), and defined as $\mathrm{Res}(Z,S) = Z - \sum_{i \in S} \alpha_i X_i$.

3.1 Approximate Submodularity of R^2

The key insight enabling our analysis is a bound on the submodularity ratio of the R^2 function. To avoid notational clutter, when we are specifically concerned with the R^2 objective defined on the variables X_i, we omit the function name in the definition of the submodularity ratio, and simply write

$$\gamma_{U,k} = \min_{L \subseteq U, S:|S| \leq k, S \cap L = \emptyset} \frac{\sum_{i \in S} (R^2_{Z,L \cup \{X_i\}} - R^2_{Z,L})}{R^2_{Z,S \cup L} - R^2_{Z,L}} = \min_{L \subseteq U, S:|S| \leq k, S \cap L = \emptyset} \frac{(\mathbf{b}^L_S)^\intercal \mathbf{b}^L_S}{(\mathbf{b}^L_S)^\intercal (C^L_S)^{-1} \mathbf{b}^L_S},$$

where C^L and $\mathbf{b}^L$ are the normalized covariance matrix and normalized covariance vector corresponding to the set $\{\mathrm{Res}(X_1,L), \mathrm{Res}(X_2,L), \ldots, \mathrm{Res}(X_n,L)\}$.

Our key lemma can now be stated as follows:

Lemma 13 $\gamma_{U,k} \geq \lambda_{\min}(C, k + |U|) \geq \lambda_{\min}(C)$.

For all our analysis in this paper, we will use $|U| = k$, and hence $\gamma_{U,k} \geq \lambda_{\min}(C, 2k)$. Thus, the smallest $2k$-sparse eigenvalue is a lower bound on this submodularity ratio; as we show later, it is often a weak lower bound.

Before proving Lemma 13, we first introduce two lemmas that relate the eigenvalues of a normalized covariance matrix with those of its submatrices.

Lemma 14 *Let C be the covariance matrix of n zero-mean random variables $X_1, X_2, \ldots, X_n$, each of which has variance at most 1. Let C_ρ be the corresponding correlation matrix of the n random variables, that is, C_ρ is the covariance matrix of the variables after they are normalized to have unit variance. Then $\lambda_{\min}(C) \leq \lambda_{\min}(C_\rho)$.*

5. Computing $\lambda_{\min}(C,k)$ is NP-hard. In Appendix A we describe how to efficiently approximate the values for some scenarios.

Proof. Since C_ρ is obtained by normalizing the variables such that they have unit variance, we get $C_\rho = D^\intercal C D$, where D is a diagonal matrix with diagonal entries $d_i = \frac{1}{\sqrt{\mathrm{Var}[X_i]}}$.

Since both C_ρ and C are positive semidefinite, we can perform Cholesky factorization to get lower-triangular matrices A_ρ and A such that $C = AA^\intercal$ and $C_\rho = A_\rho A_\rho^\intercal$. Hence $A_\rho = D^\intercal A$.

Let $\sigma_{\min}(A)$ and $\sigma_{\min}(A_\rho)$ denote the smallest singular values of A and A_ρ, respectively. Also, let $\mathbf{v}$ be the singular vector corresponding to $\sigma_{\min}(A_\rho)$. Then,

$$\|A\mathbf{v}\|_2 = \|D^{-1}A_\rho\mathbf{v}\|_2 \;\le\; \|D^{-1}\|_2\|A_\rho\mathbf{v}\|_2 \;=\; \sigma_{\min}(A_\rho)\|D^{-1}\|_2 \;\le\; \sigma_{\min}(A_\rho),$$

where the last inequality follows since

$$\|D^{-1}\|_2 \;=\; \max_i \frac{1}{d_i} \;=\; \max_i \sqrt{\mathrm{Var}[X_i]} \;\le\; 1.$$

Hence, by the Courant-Fischer theorem, $\sigma_{\min}(A) \le \sigma_{\min}(A_\rho)$, and consequently, $\lambda_{\min}(C) \le \lambda_{\min}(C_\rho)$. ∎

Lemma 15 *Let $\lambda_{\min}(C)$ be the smallest eigenvalue of the covariance matrix C of n random variables $X_1, X_2, \ldots, X_n$, and $\lambda_{\min}(C')$ be the smallest eigenvalue of the $(n-1)\times(n-1)$ covariance matrix C' corresponding to the $n-1$ random variables $\mathrm{Res}(X_1, X_n), \ldots, \mathrm{Res}(X_{n-1}, X_n)$. Then $\lambda_{\min}(C) \le \lambda_{\min}(C')$.*

Proof. Let λ_i and λ'_i denote the eigenvalues of C and C', respectively. Also, let $c'_{i,j}$ denote the entries of C'. Using the definition of the residual, we get that

$$c'_{i,j} = \mathrm{Cov}[\mathrm{Res}(X_i, X_n), \mathrm{Res}(X_j, X_n)] \;=\; c_{i,j} - \frac{c_{i,n}c_{j,n}}{c_{n,n}},$$

$$c'_{i,i} = \mathrm{Var}[\mathrm{Res}(X_i, X_n)] \;=\; c_{i,i} - \frac{c_{i,n}^2}{c_{n,n}}.$$

Defining $D = \frac{1}{c_{n,n}} \cdot [c_{1,n}, c_{2,n}, \ldots, c_{n-1,n}]^\intercal \cdot [c_{1,n}, c_{2,n}, \ldots, c_{n-1,n}]$, we can write $C_{\{1,\ldots,n-1\}} = C' + D$. To prove $\lambda_1 \le \lambda'_1$, let $\mathbf{e}' = [e'_1, \ldots, e'_{n-1}]^\intercal$ be the eigenvector of C' corresponding to the eigenvalue λ'_1, and consider the vector $\mathbf{e} = [e'_1, e'_2, \ldots, e'_{n-1}, -\frac{1}{c_{n,n}}\sum_{i=1}^{n-1} e'_i c_{i,n}]^\intercal$. Then, $C \cdot \mathbf{e} = \left[\begin{smallmatrix}\mathbf{y}\\0\end{smallmatrix}\right]$, where

$$\begin{aligned}
\mathbf{y} &= -\frac{1}{c_{n,n}}\sum_{i=1}^{n-1} e'_i c_{i,n}[c_{1,n}, c_{2,n}, \ldots, c_{n-1,n}]^\intercal + C_{\{1,\ldots,n-1\}} \cdot \mathbf{e}' \\
&= -\frac{1}{c_{n,n}}\sum_{i=1}^{n-1} e'_i c_{i,n}[c_{1,n}, c_{2,n}, \ldots, c_{n-1,n}]^\intercal + D \cdot \mathbf{e}' + C' \cdot \mathbf{e}' \\
&= C' \cdot \mathbf{e}'.
\end{aligned}$$

Thus, $C \cdot \mathbf{e} = [\lambda'_1 e'_1, \lambda'_1 e'_2, \ldots, \lambda'_1 e'_{n-1}, 0]^\intercal = \lambda'_1[e'_1, e'_2, \ldots, e'_{n-1}, 0]^\intercal \le \lambda'_1\|\mathbf{e}\|_2$, which by Rayleigh-Ritz bounds implies that $\lambda_1 \le \lambda'_1$. ∎

Using the above two lemmas, we now prove Lemma 13.

Proof of Lemma 13. Since

$$\frac{(\mathbf{b}_S^L)^\intercal (C_S^L)^{-1} \mathbf{b}_S^L}{(\mathbf{b}_S^L)^\intercal \mathbf{b}_S^L} \leq \max_{\mathbf{x}} \frac{\mathbf{x}^\intercal (C_S^L)^{-1} \mathbf{x}}{\mathbf{x}^\intercal \mathbf{x}} \;=\; \lambda_{\max}((C_S^L)^{-1}) \;=\; \frac{1}{\lambda_{\min}(C_S^L)},$$

we can use Definition 2 to obtain that

$$\gamma_{U,k} \geq \min_{(L \subseteq U, S: |S| \leq k, S \cap L = \emptyset)} \lambda_{\min}(C_S^L).$$

Next, we relate $\lambda_{\min}(C_S^L)$ with $\lambda_{\min}(C_{L \cup S})$, using repeated applications of Lemmas 14 and 15. Let $L = \{X_1, \ldots, X_\ell\}$; for each i, define $L_i = \{X_1, \ldots, X_i\}$, and let $C^{(i)}$ be the covariance matrix of the random variables $\{\mathrm{Res}(X, L \setminus L_i) \mid X \in S \cup L_i\}$, and $C_\rho^{(i)}$ the covariance matrix after normalizing all its variables to unit variance. Then, Lemma 14 implies that for each i, $\lambda_{\min}(C^{(i)}) \leq \lambda_{\min}(C_\rho^{(i)})$, and Lemma 15 shows that $\lambda_{\min}(C_\rho^{(i)}) \leq \lambda_{\min}(C^{(i-1)})$ for each $i > 0$. Combining these inequalities inductively for all i, we obtain that

$$\lambda_{\min}(C_S^L) = \lambda_{\min}(C_\rho^{(0)}) \;\geq\; \lambda_{\min}(C^{(\ell)}) \;=\; \lambda_{\min}(C_{L \cup S}) \;\geq\; \lambda_{\min}(C, |L \cup S|).$$

Finally, since $|S| \leq k$ and $L \subseteq U$, we obtain $\gamma_{U,k} \geq \lambda_{\min}(C, k + |U|)$. ■

3.2 Forward Regression

We now use our approximate submodularity framework along with the result of Lemma 13 to achieve theoretical performance bounds for Forward Regression and Orthogonal Matching Pursuit, which are widely used in practice. We also analyze the Oblivious algorithm, one of the simplest greedy algorithms for subset selection. Throughout the remainder of this section, we use $\mathrm{OPT} = \max_{S:|S|=k} R^2_{Z,S}$ to denote the optimum R^2 value achievable by any set of size k.

We begin with an analysis of Forward Regression, which is the standard algorithm used by many researchers in medical, social, and economic domains.[6]

Algorithm 2 The Forward Regression (also called Forward Selection) algorithm.

1: Initialize $S_0 = \emptyset$.
2: **for** each iteration $i + 1 = 1, 2, \ldots$ **do**
3: Let X_{i+1} be a variable maximizing $R^2_{Z, S_i \cup \{X_{i+1}\}}$, and set $S_{i+1} = S_i \cup \{X_{i+1}\}$.
4: Output S_k.

Notice that Forward Regression is exactly the special case of the general Nemhauser Greedy Algorithm (Algorithm 1) applied to the R^2 objective.

Our main result is the following theorem.

6. There is some inconsistency in the literature about naming of greedy algorithms. Forward Regression is sometimes also referred to as Orthogonal Matching Pursuit (OMP). We choose the nomenclature consistent with Miller (2002) and Tropp (2004).

Theorem 16 *The set S^{FR} selected by Forward Regression has the following approximation guarantees:*

$$\begin{aligned} R^2_{Z,S^{FR}} &\geq (1 - \mathrm{e}^{-\gamma_{S^{FR},k}}) \cdot OPT \\ &\geq (1 - \mathrm{e}^{-\lambda_{\min}(C,2k)}) \cdot OPT \\ &\geq (1 - \mathrm{e}^{-\lambda_{\min}(C,k)}) \cdot \Theta((\frac{1}{2})^{1/\lambda_{\min}(C,k)}) \cdot OPT. \end{aligned}$$

The first inequality is just an application of Theorem 6 to the R^2 objective, and the second inequality follows directly from Lemma 13 by noticing that $|S^{\mathrm{FR}}| = k$. Thus, our proof will focus on the third inequality, which relates the performance measured with respect to the smallest k-sparse eigenvalue to that measured with respect to the smallest $2k$-sparse eigenvalue. We begin with a general lemma that bounds the amount by which the R^2 value of a set and the sum of R^2 values of its elements can differ.

Lemma 17 *Let C and $\mathbf{b}$ be the covariance matrix and covariance vector corresponding to a predictor variable Z and a set S of random variables $X_1, X_2, \ldots, X_n$ that are normalized to have zero mean and unit variance. Then,*

$$\frac{1}{\lambda_{\max}(C)} \sum_{i=1}^{n} R^2_{Z,X_i} \leq R^2_{Z,\{X_1,\ldots,X_n\}} \leq \frac{1}{\gamma_{\emptyset,n}} \sum_{i=1}^{n} R^2_{Z,X_i} \leq \frac{1}{\lambda_{\min}(C)} \sum_{i=1}^{n} R^2_{Z,X_i}.$$

Proof. Let the eigenvalues of C^{-1} be $\lambda'_1 \leq \lambda'_2 \leq \ldots \leq \lambda'_n$, with corresponding orthonormal eigenvectors $\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_n}$. We write $\mathbf{b}$ in the basis $\{\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_n}\}$ as $\mathbf{b} = \sum_i \beta_i \mathbf{e_i}$. Then,

$$R^2_{Z,\{X_1,\ldots,X_n\}} = \mathbf{b}^\intercal C^{-1} \mathbf{b} = \sum_i \beta_i^2 \lambda'_i.$$

Because $\lambda'_1 \leq \lambda'_i$ for all i, we get $\lambda'_1 \sum_i \beta_i^2 \leq R^2_{Z,\{X_1,\ldots,X_n\}}$, and $\sum_i \beta_i^2 = \mathbf{b}^\intercal \mathbf{b} = \sum_i R^2_{Z,X_i}$, because the length of the vector $\mathbf{b}$ is independent of the basis it is written in. Also, by definition of the submodularity ratio, $R^2_{Z,\{X_1,\ldots,X_n\}} \leq \frac{\sum_i \beta_i^2}{\gamma_{\emptyset,n}}$. Finally, because $\lambda'_1 = \frac{1}{\lambda_{\max}(C)}$, and using Lemma 13, we obtain the result. ∎

The next lemma relates the optimal R^2 value using k elements to the optimal R^2 using $k' < k$ elements.

Lemma 18 *For each k, let $S^*_k \in \operatorname{argmax}_{|S| \leq k} R^2_{Z,S}$ be an optimal subset of at most k variables. Then, for any $k' = \Theta(k)$ such that $\frac{1}{\lambda_{\min}(C,k)} < k' < k$, we have that $R^2_{Z,S^*_{k'}} \geq R^2_{Z,S^*_k} \cdot \Theta((\frac{k'}{k})^{1/\lambda_{\min}(C,k)})$, for large enough k. In particular, $R^2_{Z,S^*_{k/2}} \geq R^2_{Z,S^*_k} \cdot \Theta((\frac{1}{2})^{1/\lambda_{\min}(C,k)})$, for large enough k.*

Proof. We first prove that $R^2_{Z,S^*_{k-1}} \geq (1 - \frac{1}{k\lambda_{\min}(C,k)}) R^2_{Z,S^*_k}$. Let $T = \mathrm{Res}(Z, S^*_k)$; then, $\mathrm{Cov}[X_i, T] = 0$ for all $X_i \in S^*_k$, and $Z = T + \sum_{X_i \in S^*_k} \alpha_i X_i$, where $\alpha = (\alpha_i) = C^{-1}_{S^*_k} \cdot \mathbf{b}_{S^*_k}$ are

the optimal regression coefficients. We write $Z' = Z - T$. For any $X_j \in S_k^*$, by definition of R^2, we have that

$$R^2_{Z',S_k^*\setminus\{X_j\}} = 1 - \frac{\alpha_j^2 \mathrm{Var}[X_j]}{\mathrm{Var}[Z']} \;=\; 1 - \frac{\alpha_j^2}{\mathrm{Var}[Z']};$$

in particular, this implies that $R^2_{Z',S_{k-1}^*} \geq 1 - \frac{\alpha_j^2}{\mathrm{Var}[Z']}$ for all $X_j \in S_k^*$.

Focus now on j minimizing α_j^2, so that $\alpha_j^2 \leq \frac{\|\alpha\|_2^2}{k}$. As in the proof of Lemma 17, by writing α in terms of an orthonormal eigenbasis of $C_{S_k^*}$, one can show that $|\alpha^\intercal C_{S_k^*}\alpha| \geq \|\alpha\|_2^2 \lambda_{\min}(C_{S_k^*})$, or $\|\alpha\|_2^2 \leq \frac{|\alpha^\intercal C_{S_k^*}\alpha|}{\lambda_{\min}(C_{S_k^*})}$. Furthermore, $\alpha^\intercal C_{S_k^*}\alpha = \mathrm{Var}[\sum_{X_i \in S_k^*} \alpha_i X_i] = \mathrm{Var}[Z']$, so $R^2_{Z',S_{k-1}^*} \geq 1 - \frac{1}{k\lambda_{\min}(C_{S_k^*})}$. Finally, by definition, $R^2_{Z',S_k^*} = 1$, so

$$\frac{R^2_{Z,S_{k-1}^*}}{R^2_{Z,S_k^*}} \geq \frac{R^2_{Z',S_{k-1}^*}}{R^2_{Z',S_k^*}} \;\geq\; 1 - \frac{1}{k\lambda_{\min}(C_{S_k^*})} \;\geq\; 1 - \frac{1}{k\lambda_{\min}(C,k)}.$$

Now, applying this inequality repeatedly, we get

$$R^2_{Z,S_{k'}^*} \geq R^2_{Z,S_k^*} \cdot \prod_{i=k'+1}^{k} (1 - \frac{1}{i\lambda_{\min}(C,i)}).$$

Let $t = \lceil 1/\lambda_{\min}(C,k) \rceil$, so that the previous bound implies $R^2_{Z,S_{k'}^*} \geq R^2_{Z,S_k^*} \cdot \prod_{i=k'+1}^{k} \frac{i-t}{i}$. Most of the terms in the product telescope, giving us a bound of $R^2_{Z,S_k^*} \cdot \prod_{i=1}^{t} \frac{k'-t+i}{k-t+i}$. Since $\prod_{i=1}^{t} \frac{k'-t+i}{k-t+i}$ converges to $(\frac{k'}{k})^t$ with increasing k (keeping t constant), we get that for large k,

$$R^2_{Z,S_{k'}^*} \geq R^2_{Z,S_k^*} \cdot \Theta((\frac{k'}{k})^t) \;\geq\; R^2_{Z,S_k^*} \cdot \Theta((\frac{k'}{k})^{1/\lambda_{\min}(C,k)}).$$

This completes the proof. ∎

Using the above lemmas, we now prove the main theorem.

Proof of Theorem 16. As mentioned earlier, the first inequality is a direct corollary of Theorem 6, obtained by replacing f with the R^2 function. The second inequality follows directly from Lemma 13 and the fact that $|S^{\mathrm{FR}}| = k$.

By applying the above result after $k/2$ iterations, we obtain $R^2_{Z,S^{\mathrm{NG}}_{k/2}} \geq (1 - \mathrm{e}^{-\lambda_{\min}(C,k)}) \cdot R^2_{Z,S^*_{k/2}}$. Now, using Lemma 18 and monotonicity of R^2, we get

$$R^2_{Z,S^{\mathrm{NG}}_k} \;\geq\; R^2_{Z,S^{\mathrm{NG}}_{k/2}} \;\geq\; (1 - \mathrm{e}^{-\lambda_{\min}(C,k)}) \cdot \Theta((\frac{1}{2})^{1/\lambda_{\min}(C,k)}) \cdot R^2_{Z,S_k^*},$$

proving the third inequality. ∎

Algorithm 3 The Orthogonal Matching Pursuit algorithm.

1: Initialize $S_0 = \emptyset$.
2: **for** each iteration $i + 1 = 1, 2, \ldots$ **do**
3: Let X_{i+1} be a variable maximizing $|\mathrm{Cov}[\mathrm{Res}(Z, S_i), X_{i+1}]|$, and set $S_{i+1} = S_i \cup \{X_{i+1}\}$.
4: Output S_k.

3.3 Orthogonal Matching Pursuit

The second greedy algorithm we analyze is Orthogonal Matching Pursuit (OMP), frequently used in signal processing domains.

By applying similar techniques as in the previous section, we can also obtain approximation bounds for OMP. We start by proving the following lemma that lower-bounds the variance of the residual of a variable.

Lemma 19 *Let A be the $(n+1) \times (n+1)$ covariance matrix of the normalized variables $Z, X_1, X_2, \ldots, X_n$. Then* $\mathrm{Var}[\mathrm{Res}(Z, \{X_1, \ldots, X_n\})] \geq \lambda_{\min}(A)$.

Proof. The matrix A is of the form $A = \begin{pmatrix} 1 & \mathbf{b}^\intercal \\ \mathbf{b} & C \end{pmatrix}$. We use $A[i, j]$ to denote the matrix obtained by removing the i^{th} row and j^{th} column of A, and similarly for C. Recalling that the (i, j) entry of C^{-1} is $\frac{(-1)^{i+j} \det(C[i,j])}{\det(C)}$, and developing the determinant of A by the first row and column, we can write

$$\begin{aligned}
\det(A) &= \sum_{j=1}^{n+1} (-1)^{1+j} a_{1,j} \det(A[1, j]) \\
&= \det(C) + \sum_{j=1}^{n} (-1)^j b_j \det(A[1, j+1]) \\
&= \det(C) + \sum_{j=1}^{n} (-1)^j b_j \sum_{i=1}^{n} (-1)^{i+1} b_i \det(C[i, j]) \\
&= \det(C) - \sum_{j=1}^{n} \sum_{i=1}^{n} (-1)^{i+j} b_i b_j \det(C[i, j]) \\
&= \det(C)(1 - \mathbf{b}^\intercal C^{-1} \mathbf{b}).
\end{aligned}$$

Therefore, using that $\mathrm{Var}[Z] = 1$,

$$\mathrm{Var}[\mathrm{Res}(Z, \{X_1, \ldots, X_n\})] \;=\; \mathrm{Var}[Z] - \mathbf{b}^\intercal C^{-1} \mathbf{b} \;=\; \frac{\det(A)}{\det(C)}.$$

Because $\det(A) = \prod_{i=1}^{n+1} \lambda_i^A$ and $\det(C) = \prod_{i=1}^{n} \lambda_i^C$, and $\lambda_1^A \leq \lambda_1^C \leq \lambda_2^A \leq \lambda_2^C \leq \ldots \leq \lambda_{n+1}^A$ by the eigenvalue interlacing theorem, we get that $\frac{\det(A)}{\det(C)} \geq \lambda_1^A$, proving the lemma. ∎

The above lemma, along with an analysis similar to the proof of Theorem 16, can be used to prove the following approximation bounds for OMP:

Theorem 20 *The set S^{OMP} selected by orthogonal matching pursuit has the following approximation guarantees:*

$$\begin{aligned} R^2_{Z,S^{OMP}} &\geq (1 - \mathrm{e}^{-(\gamma_{S^{OMP},k} \cdot \lambda_{\min}(C,2k))}) \cdot OPT \\ &\geq (1 - \mathrm{e}^{-\lambda_{\min}(C,2k)^2}) \cdot OPT \\ &\geq (1 - \mathrm{e}^{-\lambda_{\min}(C,k)^2}) \cdot \Theta((\frac{1}{2})^{1/\lambda_{\min}(C,k)}) \cdot OPT. \end{aligned}$$

Proof. We begin by proving the first inequality. Using notation similar to that in the proof of Theorem 16, we let S^*_k be the optimum set of k variables, S^{OMP}_i the set of variables chosen by OMP in the first i iterations, and $S_i = S^*_k \setminus S^{\mathrm{OMP}}_i$. For each $X_j \in S_i$, let $X'_j = \mathrm{Res}(X_j, S^{\mathrm{OMP}}_i)$ be the residual of X_j conditioned on S^{OMP}_i, and write $S'_i = \{X'_j \mid X_j \in S_i\}$.

Consider some iteration $i+1$ of OMP. We will show that at least one of the X'_i is a good candidate in this iteration. Let ℓ maximize R^2_{Z,X'_ℓ}, i.e., $\ell \in \mathrm{argmax}_{(j:X'_j \in S'_i)} R^2_{Z,X'_j}$. By Lemma 19,

$$\mathrm{Var}[X'_\ell] \geq \lambda_{\min}(C_{S^i_G \cup \{X'_\ell\}}) \;\geq\; \lambda_{\min}(C, 2k).$$

The OMP algorithm chooses a variable X_m to add which maximizes $|\mathrm{Cov}[\mathrm{Res}(Z, S^i_G), X_m]|$. Thus, X_m maximizes

$$\mathrm{Cov}[\mathrm{Res}(Z, S^i_G), X_m]^2 = \mathrm{Cov}[Z, \mathrm{Res}(X_m, S^i_G)]^2 \;=\; R^2_{Z,\mathrm{Res}(X_m,S^i_G)} \cdot \mathrm{Var}[\mathrm{Res}(X_m, S^i_G)].$$

In particular, this implies

$$\begin{aligned} R^2_{Z,\mathrm{Res}(X_m,S^i_G)} &\geq R^2_{Z,X'_\ell} \cdot \frac{\mathrm{Var}[X'_\ell]}{\mathrm{Var}[\mathrm{Res}(X_m, S^i_G)]} \\ &\geq R^2_{Z,X'_\ell} \cdot \frac{\lambda_{\min}(C,2k)}{\mathrm{Var}[\mathrm{Res}(X_m, S^i_G)]} \;\geq\; R^2_{Z,X'_\ell} \cdot \lambda_{\min}(C,2k), \end{aligned}$$

because $\mathrm{Var}[\mathrm{Res}(X_m, S^i_G)] \leq 1$. As in the proof of Theorem 6, $R^2_{Z,X'_\ell} \geq \frac{\gamma_{S^{\mathrm{OMP}},k}}{k} \cdot R^2_{Z,S'_i}$, so $R^2_{Z,\mathrm{Res}(X_m,S^i_G)} \geq R^2_{Z,S'_i} \cdot \frac{\lambda_{\min}(C,2k) \cdot \gamma_{S^{\mathrm{OMP}},k}}{k}$. With the same definition of $A(i)$ as in the proof of Theorem 6, we get that $A(i+1) \geq \frac{\lambda_{\min}(C,2k)\gamma_{S^{\mathrm{OMP}},k}}{k} \cdot (\mathrm{OPT} - \sum_{j=1}^{i} A(j))$. An inductive proof now shows that

$$R^2_{Z,S_G} \;=\; \sum_{i=1}^{k} A(i) \;\geq\; (1 - \mathrm{e}^{-\lambda_{\min}(C,2k) \cdot \gamma_{S^{\mathrm{OMP}},k}}) \cdot R^2_{Z,S^*_k}.$$

The proofs of the other two inequalities follow the same pattern as the proof for Forward Regression. ∎

3.4 Oblivious Algorithm

As a baseline, we also consider a greedy algorithm which completely ignores C and simply selects the k variables individually most correlated with Z.

Lemma 17 immediately implies a simple bound for the oblivious algorithm:

Algorithm 4 The oblivious algorithm.

1: Sort the X_i by non-increasing b_i values.
2: Return $\{X_1, X_2, \ldots, X_K\}$.

Theorem 21 *The set S^{OBL} selected by the oblivious algorithm has the following approximation guarantees:*

$$R^2_{Z,S^{OBL}} \;\geq\; \frac{\gamma_{\emptyset,k}}{\lambda_{\max}(C,k)} \cdot OPT \;\geq\; \frac{\lambda_{\min}(C,k)}{\lambda_{\max}(C,k)} \cdot OPT.$$

Proof. Let S be the set chosen by the oblivious algorithm, and S^*_k the optimum set of k variables. By definition of the oblivious algorithm, $\sum_{i\in S} R^2_{Z,X_i} \geq \sum_{i\in S^*_k} R^2_{Z,X_i}$, so using Lemma 17, we obtain that

$$R^2_{Z,S} \geq \frac{\sum_{i\in S} R^2_{Z,X_i}}{\lambda_{\max}(C,k)} \;\geq\; \frac{\sum_{i\in S^*_k} R^2_{Z,X_i}}{\lambda_{\max}(C,k)} \;\geq\; \frac{\gamma_{\emptyset,k}}{\lambda_{\max}(C,k)} R^2_{Z,S^*_k}.$$

The second inequality of the theorem follows directly from Lemma 13. ■

4. Dictionary Selection Bounds

To demonstrate the wider applicability of the approximate submodularity framework, we next obtain a tighter analysis for two greedy algorithms for the dictionary selection problem, introduced by Krause and Cevher (2010).

The Dictionary Selection problem generalizes the Subset Selection problem by considering s predictor variables $Z_1, Z_2, \ldots, Z_s$. The goal is to select a dictionary D of d observation variables, to optimize the average R^2 fit for the Z_i using at most k vectors from D for each. Formally, the Dictionary Selection problem is defined as follows:

Definition 22 (Dictionary Selection) *Given all pairwise covariances among the Z_j and X_i variables, as well as parameters d and k, find a set D of at most d variables from $\{X_1, \ldots, X_n\}$ maximizing*

$$F(D) = \sum_{j=1}^{s} \max_{S\subset D, |S|=k} R^2_{Z_j,S}.$$

4.1 The Algorithm SDS_{MA}

The SDS_{MA} algorithm generalizes the oblivious greedy algorithm to the problem of Dictionary Selection. It replaces the $R^2_{Z_j,S}$ term in Definition 22 with its modular approximation $f(Z_j, S) = \sum_{i\in S} R^2_{Z_j,X_i}$. Thus, it greedily tries to maximize the function $\hat{F}(D) = \sum_{j=1}^{s} \max_{S\subset D,|S|=k} f(Z_j, S)$, over sets D of size at most d; the inner maximum can be computed efficiently using the oblivious algorithm.

Using Lemma 17, we obtain the following multiplicative approximation guarantee for SDS_{MA}:

Algorithm 5 The SDS_{MA} algorithm for dictionary selection.

Initialize $D_0 = \emptyset$.
for each iteration $i + 1 = 1, 2, \ldots$ **do**
  Let X_{i+1} be a variable maximizing $\hat{F}(D \cup \{X_m\})$, and set $S_{i+1} = S_i \cup \{X_{i+1}\}$.
Output D_d.

Theorem 23 *Let D^{MA} be the dictionary selected by the SDS_{MA} algorithm, and D^* the optimum dictionary of size $|D| \leq d$, with respect to the objective $F(D)$ from Definition 22. Then,*

$$F(D^{MA}) \geq \frac{\gamma_{\emptyset,k}}{\lambda_{\max}(C,k)}(1 - \frac{1}{\mathrm{e}}) \cdot F(D^*) \;\geq\; \frac{\lambda_{\min}(C,k)}{\lambda_{\max}(C,k)}(1 - \frac{1}{\mathrm{e}}) \cdot F(D^*).$$

Proof. Let $\hat{D}$ be a dictionary of size d maximizing $\hat{F}(D)$. Because $f(Z_j, S)$ is monotone and modular in S, $\hat{F}$ is a monotone, submodular function. Hence, using the submodularity results of Nemhauser et al. (1978) and the optimality of $\hat{D}$ for $\hat{F}$,

$$\hat{F}(D^{\mathrm{MA}}) \geq \hat{F}(\hat{D}) \cdot (1 - \frac{1}{\mathrm{e}}) \;\geq\; \hat{F}(D^*) \cdot (1 - \frac{1}{\mathrm{e}}).$$

Now, by applying Lemma 17 for each Z_j, it is easy to show that $\hat{F}(D^*) \geq \gamma_{\emptyset,k} \cdot F(D^*)$, and similarly $\hat{F}(D^{\mathrm{MA}}) \leq \lambda_{\max}(C,k) \cdot F(D^{\mathrm{MA}})$. Thus we get $F(D^{\mathrm{MA}}) \geq \frac{\gamma_{\emptyset,k}}{\lambda_{\max}(C,k)}(1 - \frac{1}{\mathrm{e}})F(D^*)$.

The second part now follows from Lemma 13. ■

Note that these bounds significantly improve the previous additive approximation guarantee obtained by Krause and Cevher (2010): $F(D^{\mathrm{MA}}) \geq (1 - \frac{1}{\mathrm{e}}) \cdot F(D^*) - (2 - \frac{1}{\mathrm{e}}) \cdot k \cdot \mu(C)$. In particular, when $\mu(C) > \Theta(1/k)$, i.e., even just one pair of variables has moderate correlation, the approximation guarantee of Krause and Cevher becomes trivial.

4.2 The Algorithm $\mathrm{SDS_{OMP}}$

We also obtain a multiplicative approximation guarantee for the greedy $\mathrm{SDS_{OMP}}$ algorithm, introduced by Krause and Cevher for dictionary selection. Our bounds for $\mathrm{SDS_{OMP}}$ are much stronger than the additive bounds obtained by Krause and Cevher. However, for both our results and theirs, the performance guarantees for $\mathrm{SDS_{OMP}}$ are much weaker than those for $\mathrm{SDS_{MA}}$.

The $\mathrm{SDS_{OMP}}$ algorithm generalizes the Orthogonal Matching Pursuit algorithm for subset selection to the problem of dictionary selection. In each iteration, it adds a new element to the currently selected dictionary by using Orthogonal Matching Pursuit to approximate the estimation of $\max_{|S|=k} R^2_{Z_j,S}$.

We now show how to obtain a multiplicative approximation guarantee for $\mathrm{SDS_{OMP}}$. The following definitions are key to our analysis; the first two are from Definition 22 and

Algorithm 6 The SDS_{OMP} algorithm for dictionary selection.

1: Initialize $D_0 = \emptyset$.
2: **for** each iteration $i+1 = 1, 2, \ldots$ **do**
3: Let X_{i+1} be a variable maximizing $\sum_{j=1}^{s} R^2_{Z_j, S_{\mathrm{OMP}}(D_i \cup \{X_{i+1}\}, Z_j, k)}$ where $S_{\mathrm{OMP}}(D, Z, k)$ denotes the set selected by Orthogonal Matching Pursuit for predicting Z using k variables from D.
4: Set $S_{i+1} = S_i \cup \{X_{i+1}\}$.
5: Output D_d.

Theorem 23.

$$F(D) = \sum_{j=1}^{s} \max_{S \subset D, |S|=k} R^2_{Z_j, S},$$

$$\hat{F}(D) = \sum_{j=1}^{s} \max_{S \subset D, |S|=k} f(Z_j, S),$$

$$\tilde{F}(D) = \sum_{j=1}^{s} R^2_{Z_j, S_{\mathrm{OMP}}(D, Z_j, k)}.$$

We first prove the following lemma about approximating the function $\hat{F}(D)$ by $\tilde{F}(D)$:

Lemma 24 *For any set D, we have that*

$$\frac{(1 - \mathrm{e}^{-\lambda_{\min}(C,2k)^2})}{\lambda_{\max}(C,k)} \cdot \hat{F}(D) \leq \tilde{F}(D) \ \leq \ \frac{\hat{F}(D)}{\gamma_{\emptyset,k}}.$$

Proof. Using Theorem 20 and Lemma 17 and summing up over all the Z_j terms, we obtain that

$$\tilde{F}(D) \geq (1 - \mathrm{e}^{-\lambda_{\min}(C,2k)^2}) \cdot F(D) \ \geq \ (1 - \mathrm{e}^{-\lambda_{\min}(C,2k)^2}) \frac{\hat{F}(D)}{\lambda_{\max}(C,k)}.$$

Similarly, using Lemma 17 and the fact that $\max_{S \subset D, |S|=k} R^2_{Z_j,S} \geq R^2_{Z_j, S_{OMP}(D, Z_j, k)}$, we have

$$\hat{F}(D) \geq \gamma_{\emptyset,k} \cdot F(D) \ \geq \ \gamma_{\emptyset,k} \cdot \tilde{F}(D). \ \blacksquare$$

Using the above lemma, we now prove the following bound for $\mathrm{SDS}_{\mathrm{OMP}}$:

Theorem 25 *Let D^{OMP} be the dictionary selected by the SDS_{OMP} algorithm, and D^* the optimum dictionary of size $|D| \leq d$, with respect to the objective $F(D)$ from Definition 22. Then,*

$$F(D^{OMP}) \geq F(D^*) \cdot \frac{\gamma_{\emptyset,k}}{\lambda_{\max}(C,k)} \cdot \frac{(1 - \mathrm{e}^{-(p \cdot \gamma_{\emptyset,k})})}{d - d \cdot p \cdot \gamma_{\emptyset,k} + 1} \ \geq \ F(D^*) \cdot \frac{\lambda_{\min}(C,k)}{\lambda_{\max}(C,k)} \cdot \frac{(1 - \mathrm{e}^{-(p \cdot \gamma_{\emptyset,k})})}{d - d \cdot p \cdot \gamma_{\emptyset,k} + 1},$$

where $p = \frac{1}{\lambda_{\max}(C,k)} \cdot (1 - \mathrm{e}^{-\lambda_{\min}(C,2k)^2})$.

Proof. Let $\hat{D}$ be the dictionary of size d that maximizes $\hat{F}(D)$. We first prove that $\hat{F}(D^{\mathrm{OMP}})$ is a good approximation to $\hat{F}(\hat{D})$.

Let S_i^{NG} be the variables chosen by $\mathrm{SDS}_{\mathrm{OMP}}$ after i iterations. Define $S_i = \hat{D} \setminus S_i^{\mathrm{NG}}$. By monotonicity of $\hat{F}$, we have that $\hat{F}(S_i \cup S_i^{\mathrm{NG}}) \geq \hat{F}(\hat{D})$.

Let $\hat{X} \in S_i$ be the variable maximizing $\hat{F}(S_i^{\mathrm{NG}} \cup \{\hat{X}\})$, and similarly $\tilde{X} \in S_i$ be the variable maximizing $\tilde{F}(S_i^{\mathrm{NG}} \cup \{\tilde{X}\})$.

Since $\hat{F}$ is a submodular function, it is easy to show (using an argument similar to the proof of Theorem 16) that $\hat{F}(S_i^{\mathrm{NG}} \cup \{\hat{X}\}) - \hat{F}(S_i^{\mathrm{NG}}) \geq \frac{\hat{F}(\hat{D}) - \hat{F}(S_i^{\mathrm{NG}})}{d}$.

Now, using Lemma 24 above, and the optimality of $\tilde{X}$ for $\tilde{F}(S_i^{\mathrm{NG}} \cup \{\tilde{X}\})$, we obtain that

$$\frac{1}{\gamma_{\emptyset,k}} \cdot \hat{F}(S_i^{\mathrm{NG}} \cup \{\tilde{X}\}) \geq \tilde{F}(S_i^{\mathrm{NG}} \cup \{\tilde{X}\}) \;\geq\; \tilde{F}(S_i^{\mathrm{NG}} \cup \{\hat{X}\}) \;\geq\; p \cdot \hat{F}(S_i^{\mathrm{NG}} \cup \{\hat{X}\}).$$

Thus, $\hat{F}(S_i^{\mathrm{NG}} \cup \{\tilde{X}\}) \geq p \cdot \gamma_{\emptyset,k} \cdot \hat{F}(S_i^{\mathrm{NG}} \cup \{\hat{X}\})$, or

$$\hat{F}(S_i^{\mathrm{NG}} \cup \{\tilde{X}\}) - \hat{F}(S_i^{\mathrm{NG}}) \geq p \cdot \gamma_{\emptyset,k} \cdot (\hat{F}(S_i^{\mathrm{NG}} \cup \{\hat{X}\}) - \hat{F}(S_i^{\mathrm{NG}})) - (1 - p \cdot \gamma_{\emptyset,k})\hat{F}(S_i^{\mathrm{NG}}).$$

Define $A(i) = \hat{F}(S_i^{\mathrm{NG}}) - \hat{F}(S_{i-1}^{\mathrm{NG}})$ to be the gain, with respect to $\hat{F}$, obtained from the variable chosen by $\mathrm{SDS}_{\mathrm{OMP}}$ in iteration i. Then $\hat{F}(D^{\mathrm{OMP}}) = \sum_{i=1}^{d} A(i)$. From the preceding paragraphs, we obtain

$$A(i+1) \geq \frac{p \cdot \gamma_{\emptyset,k}}{d} \cdot (\hat{F}(\hat{D}) - (1 + \frac{d}{p \cdot \gamma_{\emptyset,k}} - d) \sum_{j=1}^{i} A(j)).$$

Since the above inequality holds for each iteration $i = 1, 2, \ldots, d$, a simple inductive proof shows that

$$\hat{F}(\hat{D}) - \sum_{i=1}^{d} A(i) \leq \hat{F}(\hat{D}) \cdot (1 - \frac{p\gamma_{\emptyset,k}}{d})^d + (d - dp\gamma_{\emptyset,k}) \cdot \sum_{i=1}^{d} A(i).$$

Rearranging the terms and simplifying, we get that

$$\hat{F}(D^{\mathrm{OMP}}) = \sum_{i=1}^{d} A(i) \;\geq\; \hat{F}(\hat{D}) \cdot \frac{(1 - \mathrm{e}^{-(p \cdot \gamma_{\emptyset,k})})}{d - dp\gamma_{\emptyset,k} + 1} \;\geq\; \hat{F}(D^*) \cdot \frac{(1 - \mathrm{e}^{-(p \cdot \gamma_{\emptyset,k})})}{d - dp\gamma_{\emptyset,k} + 1},$$

where the last inequality is due to the optimality of $\hat{D}$ for $\hat{F}$.

Now, using Lemma 17 for each Z_j term, it can be easily seen that $\hat{F}(D^*) \geq \gamma_{\emptyset,k} \cdot F(D^*)$. Similarly, using Lemma 3.3 on the set D^{OMP}, we have $F(D^{\mathrm{OMP}}) \geq \frac{1}{\lambda_{\max}(C,k)} \cdot \hat{F}(D^{\mathrm{OMP}})$.

Using the above inequalities, we therefore get the desired bound

$$F(D^{\mathrm{OMP}}) \geq F(D^*) \cdot \frac{\gamma_{\emptyset,k}}{\lambda_{\max}(C,k)} \cdot \frac{(1 - \mathrm{e}^{-(p \cdot \gamma_{\emptyset,k})})}{d - d \cdot p \cdot \gamma_{\emptyset,k} + 1}.$$

The second inequality of the Theorem now follows directly from Lemma 13. ∎

5. Experiments

In this section, we evaluate Forward Regression (FR) and OMP empirically, on two real-world and one synthetic data set. We compare the two algorithms against an optimal solution (OPT), computed using exhaustive search, the oblivious greedy algorithm (OBL), and the L1-regularization/Lasso (L1) algorithm (in the implementation of Koh et al. (2008)). Beyond the algorithms' performance, we also compute the various spectral parameters from which we can derive lower bounds. Specifically, these are

1. the submodularity ratio: $\gamma_{S^{\mathrm{FR}},k}$, where S^{FR} is the subset selected by forward regression.
2. the smallest sparse eigenvalues $\lambda_{\min}(C,k)$ and $\lambda_{\min}(C,2k)$. (In some cases, computing $\lambda_{\min}(C,2k)$ was not computationally feasible due to the problem size.)
3. the sparse inverse condition number $\kappa(C,k)^{-1}$. As mentioned earlier, the sparse inverse condition number $\kappa(C,k)$ is strongly related to the Restricted Isometry Property in (Candès et al., 2005).
4. the smallest eigenvalue $\lambda_{\min}(C) = \lambda_{\min}(C,n)$ of the entire covariance matrix.

The aim of our experiments is twofold: First, we wish to evaluate which among the submodular and spectral parameters are good predictors of the performance of greedy algorithms in practice. Second, we wish to highlight how the theoretical bounds for subset selection algorithms reflect on their actual performance. Our analytical results predict that Forward Regression should outperform OMP, which in turn outperforms Oblivious. For Lasso, it is not known whether strong multiplicative bounds, like the ones we proved for Forward Regression or OMP, can be obtained.

5.1 Data Sets

Because several of the spectral parameters (as well as the optimum solution) are **NP**-hard to compute, we restrict our experiments to data sets with $n \leq 30$ features, from which $k \leq 8$ are to be selected. We stress that the greedy algorithms themselves are very efficient, and the restriction on data set sizes is only intended to allow for an adequate evaluation of the results.

Each data set contains $m > n$ samples, from which we compute the empirical covariance matrix (analogous to the Gram matrix in sparse approximation) between all observation variables and the predictor variable; we then normalize it to obtain C and $\mathbf{b}$. We evaluate the performance of all algorithms in terms of their R^2 fit; thus, we implicitly treat C and $\mathbf{b}$ as the ground truth, and also do not separate the data sets into training and test cases.

Our data sets are the *Boston Housing Data*, a data set of *World Bank Development Indicators*, and a synthetic data set generated from a distribution similar to the one used by Zhang (2008). The *Boston Housing Data* (available from the UCI Machine Learning Repository) is a small data set frequently used to evaluate ML algorithms. It comprises $n = 15$ features (such as crime rate, property tax rates, etc.) and $m = 516$ observations. Our goal is to predict housing prices from these features. The *World Bank Data* (available from `http://databank.worldbank.org`) contains an extensive list of socio-economic and

health indicators of development, for many countries and over several years. We choose a subset of $n = 29$ indicators for the years 2005 and 2006, such that the values for all of the $m = 65$ countries are known for each indicator. (The data set does not contain all indicators for each country.) We choose to predict the average life expectancy for those countries.

To perform tests in a controlled fashion, we also generate random instances from a known distribution similar to one used by Zhang (2008): There are $n = 29$ features, and $m = 100$ data points are generated from a joint Gaussian distribution with moderately high correlations of 0.6. The target vector is obtained by generating coefficients uniformly from 0 to 10 along each dimension, and adding noise with variance $\sigma^2 = 0.1$. Notice that the target vector is not truly sparse. As for the other two data sets, the covariances are then taken to be the empirical ones of the generated data. The plots we show are the average R^2 values for 20 independent runs of the experiment.

5.2 Results

We run the different subset selection algorithms for values of k from 2 through 8, and plot the R^2 values for the selected sets. When including all of the features, the R^2 value is close to 1 in all data sets, implying that nearly all of the variance in the function to be predicted can be explained by the features.

Figures 1, 3 and 5 show the results for the three data sets. The main insight is that on all data sets, Forward Regression performs optimally or near-optimally, and OMP is only slightly worse. This is despite the fact that (as we discuss shortly) the spectral properties would not necessarily predict such near-optimal performance. Lasso performs somewhat worse on all data sets, and, not surprisingly, the baseline oblivious algorithm performs even worse. The last fact implies that the optimal solution is non-trivial in that it must account for correlation between the observation variables. The order of performance of the greedy algorithms match the order of the strength of the theoretical bounds we derived for them.

On the World Bank data (Figure 3), all algorithms perform quite well with just 2–3 features already. The main reason is that adolescent birth rate is by itself highly predictive of life expectancy, so the first feature selected by all algorithms already contributes high R^2 value.

Figures 2, 4 and 6 show the different spectral quantities for the data sets, for varying values of k. Both of the real-world data sets are nearly singular, as evidenced by the small $\lambda_{\min}(C)$ values. In fact, the near-singularities manifest themselves for small values of k already; in particular, since $\lambda_{\min}(C, 2)$ is already small, we observe that there are pairs of highly correlated observations variables in the data sets. Thus, the bounds on approximation we would obtain by considering merely $\lambda_{\min}(C, k)$ or $\lambda_{\min}(C, 2k)$ would be quite weak. Notice, however, that they are still quite a bit stronger than the inverse condition number $\kappa(C, k)^{-1}$: this bound — which is closely related to the RIP property frequently at the center of sparse approximation analysis — takes on much smaller values, and thus would be an even looser bound than the eigenvalues.

On the other hand, the submodularity ratios $\gamma_{S^{\text{FR}},k}$ for all the data sets are much larger than the other spectral quantities (almost 5 times larger, on average, than the corresponding $\lambda_{\min}(C)$ values). Notice that unlike the other quantities, the submodularity ratios are not

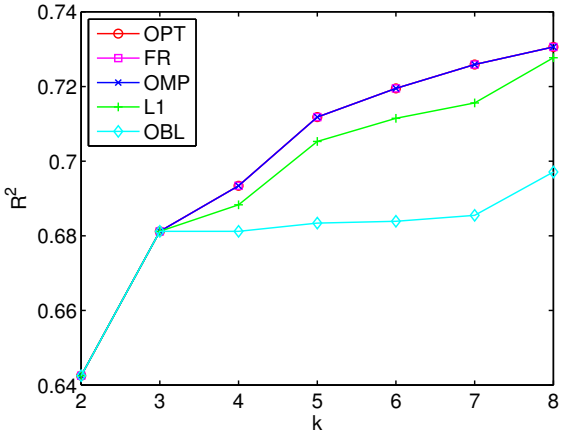

Figure 1: Boston Housing R^2

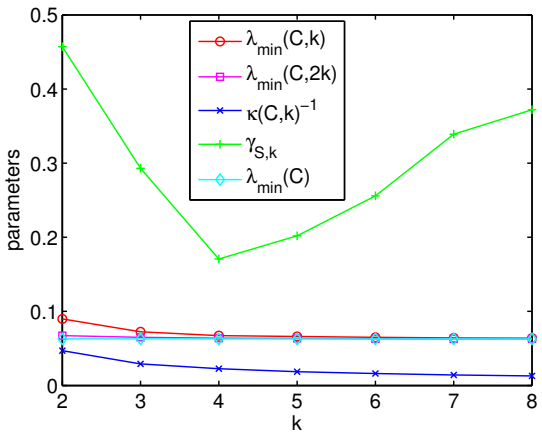

Figure 2: Boston Housing parameters

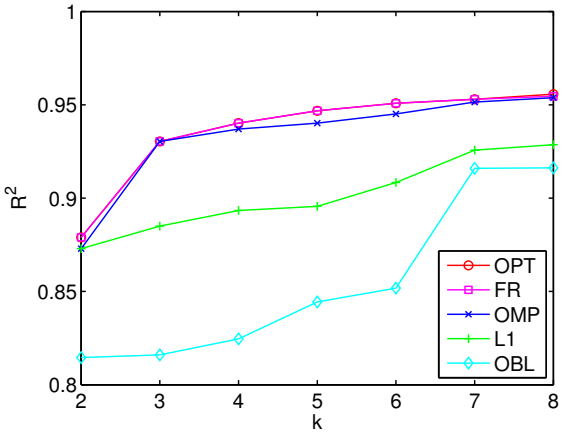

Figure 3: World Bank R^2

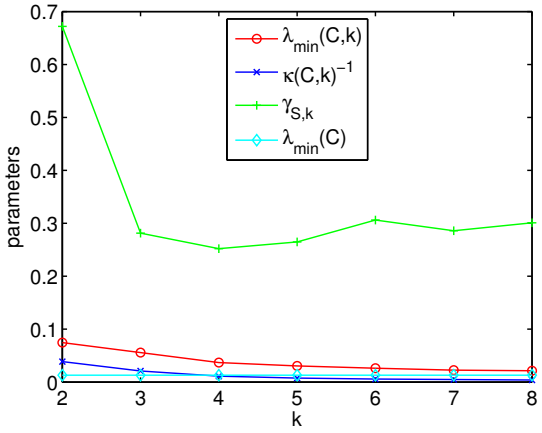

Figure 4: World Bank parameters

monotonically decreasing in k — this is due to the dependency of $\gamma_{S^{\mathrm{FR}},k}$ on the set S^{FR}, which is different for every k.

The discrepancy between the small values of the eigenvalues and the good performance of all algorithms shows that bounds based solely on eigenvalues can sometimes be loose. Significantly better bounds are obtained from the submodularity ratio $\gamma_{S^{\mathrm{FR}},k}$, which takes on values above 0.2, and significantly larger in many cases. While not entirely sufficient to explain the performance of the greedy algorithms, it shows that the near-singularities of C do not align unfavorably with $\mathbf{b}$, and thus do not provide an opportunity for strong supermodular behavior that adversely affects greedy algorithms.

The synthetic data set we generated is somewhat further from singular, with $\lambda_{\min}(C) \approx 0.11$. However, the same patterns persist: the simple eigenvalue based bounds, while some-

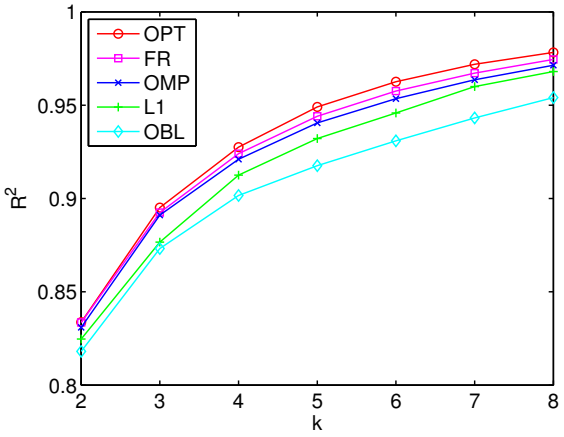

Figure 5: Synthetic Data R^2

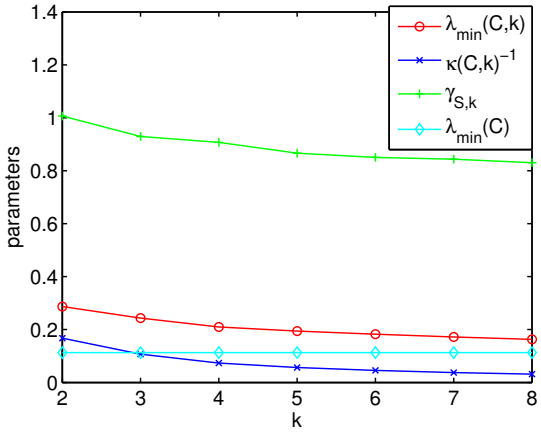

Figure 6: Synthetic Data parameters

what larger for small k, still do not fully predict the performance of greedy algorithms, whereas the submodularity ratio here is close to 1 for all values of k. This shows that the near-singularities do not at all provide the possibility of strongly supermodular benefits of sets of variables. Indeed, the plot of R^2 values on the synthetic data is concave, an indicator of submodular behavior of the function.

The above observations suggest that bounds based on the submodularity ratio are better predictors of the performance of greedy algorithms, followed by bounds based on the sparse eigenvalues, and finally those based on the condition number or RIP property.

5.3 Narrowing the gap between theory and practice

Our theoretical bounds, though much stronger than previous results, still do not fully predict the observed near-optimal performance of Forward Regression and OMP on the real-world datasets. In particular, for Forward Regression, even though the submodularity ratio is less than 0.4 for most cases, implying a theoretical guarantee of roughly $1 - e^{-0.4} \approx 33\%$, the algorithm still achieves near-optimal performance. While gaps between worst-case bounds and practical performance are commonplace in algorithmic analysis, they also suggest that there is scope for further improving the analysis, by looking at more fine-grained parameters.

Indeed, a slightly more careful analysis of the proof of Theorem 16 and our definition of the submodularity ratio reveals that we do not really need to calculate the submodularity ratio over all sets S of size k while analyzing the greedy steps of Forward Regression. We can ignore sets S whose submodularity ratio is low, but whose marginal contribution to the current R^2 is only a small fraction (say, at most ϵ). This is because the proof of Theorem 16 shows that for each iteration $i+1$, we only need to consider the submodularity ratio for the set $S_i = S_k^* \setminus S_i^{\mathrm{NG}}$, where S_i^{NG} is the set selected by the greedy algorithm after i iterations, and S_k^* is the optimal k-subset. Thus, if $R^2_{Z,S_i \cup S_i^{\mathrm{NG}}} \leq (1+\epsilon) \cdot R^2_{Z,S_i^{\mathrm{NG}}}$, then the currently selected set must already be within a factor $\frac{1}{1+\epsilon}$ of optimal.

By carefully pruning such sets (using $\epsilon = 0.2$) while calculating the submodularity ratio, we see that the resulting values of $\gamma_{S^{\text{FR}},k}$ are much higher (more than 0.8), thus significantly reducing the gap between the theoretical bounds and experimental results. Table 1 shows the values of $\gamma_{S^{\text{FR}},k}$ obtained using this method.

The results suggest an interesting direction for future work: namely, to characterize for which sets the submodular behavior of R^2 really matters.

Data Set	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$
Boston	0.9	0.91	1.02	1.21	1.36	1.54	1.74
World Bank	0.8	0.81	0.81	0.81	0.94	1.19	1.40

Table 1: Improved estimates for submodularity ratio

6. Discussion and Concluding Remarks

In this paper, we defined a notion of approximate submodularity. We showed that it naturally captures the performance degradation of the greedy algorithm. As a concrete application of the framework, we connected the submodularity ratio with spectral parameters of the covariance matrix to obtain the strongest known approximation guarantees for the Forward Selection and Orthogonal Matching Pursuit algorithms for regression. As a second example, we gave improved approximation guarantees for known greedy algorithms for dictionary selection.

We believe that our techniques for analyzing greedy algorithms using a notion of "approximate submodularity" are not specific to subset selection and dictionary selection, and could also be used to analyze other problems in compressed sensing and sparse recovery. A natural further direction is hence to identify other applications of the approximate submodularity technique.

While approximation guarantees for the greedy algorithm are perhaps the most widely used consequence of submodularity, they are far from the only one. Some other useful consequences include the following:

1. In *valid utility games* (Vetta, 2002), where utility functions are essentially submodular and interact with each other in certain ways, equilibria always achieve high social welfare.

2. A monotone submodular function can be approximately maximized subject to a Knapsack constraint (Sviridenko, 2004), Matroid constraint (Vondrák, 2008) or combinations thereof (e.g., (Chekuri et al., 2011)).

3. If the function f is submodular, but not necessarily monotone, it can be approximately maximized, with or without a cardinality constraint. Without a cardinality constraint, it can also be exactly minimized.

It would be desirable to verify whether some of these results gracefully degrade when the submodularity ratio is bounded away from 0. The third property (optimization of non-monotone submodular functions) seems unlikely to carry over, as our definition was targeted

at monotone submodular functions. This raises the natural question of whether there is a more general definition of approximate submodularity that retains the positive results of the present work while also yielding an analogue to some or all of the above properties.

Our bicriteria approximation guarantees, trading off a maximization of coverage against a minimization of cost, could be generalized to more general constraints. For instance, Iyer and Bilmes (Iyer and Bilmes, 2013) give bicriteria approximation guarantees for maximizing a submodular function subject to a submodular cost constraint, or minimizing a submodular function subject to a submodular coverage constraint. It is natural to ask whether similar guarantees can be obtained for approximately submodular functions.

As discussed in Remark 4, it is open how well one can approximate the submodularity ratio of a given function f in general; being able to do so would allow one to obtain approximation guarantees at least for specific instances. Alternatively, it may be possible to establish approximation hardness results for computing the submodularity ratio.

The approximation guarantees of the greedy algorithm are worst when the covariance matrix is singular, or close to singular. When the covariance matrix is estimated from data (rather than explicitly given), the natural variance in data generated from joint distributions may keep it from being too close to singular. A detailed investigation would constitute an interesting direction for future work, though to be useful, it would have to provide a lower bound of $\omega(1/\log n)$ on the smallest (sparse) eigenvalue.

Acknowledgments

We would like to thank Andreas Krause, Fei Sha and several anonymous referees for their helpful feedback. This work was supported in part by NSF grant 0540420 (DDDAS-TMRP).

References

Wenruo Bai and Jeffrey A. Bilmes. Greed is still good: Maximizing monotone submodular+supermodular functions, 2018. https://arxiv.org/pdf/1801.07413.pdf.

Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6): 1740–1766, 2011.

Emmanuel J. Candès and Terence Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

Emmanuel J. Candès, Justin Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59:1207–1223, 2005.

Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *Proc. 43rd ACM Symp. on Theory of Computing*, pages 783–792, 2011.

Flavio Chierichetti, Abhimanyu Das, Anirban Dasgupta, and Ravi Kumar. Approximate modularity. In *Proc. 56th IEEE Symp. on Foundations of Computer Science*, pages 1143–1162, 2015.

Michele Conforti and Gérard Cornuéjols. Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the rado-edmonds theorem. *Discrete Applied Mathematics*, 7:251–274, 1984.

Abhimanyu Das and David Kempe. Algorithms for subset selection in linear regression. In *Proc. 40th ACM Symp. on Theory of Computing*, pages 97–108, 2008.

Abhimanyu Das, Anirban Dasgupta, and Ravi Kumar. Selecting diverse features via spectral regularization. In *Proc. 26th Advances in Neural Information Processing Systems*, pages 1592–1600, 2012.

George M. Diekhoff. *Statistics for the Social and Behavioral Sciences*. Brown & Benchmark, 2002.

David L. Donoho. For most large underdetermined systems of linear equations, the minimal 11-norm near-solution approximates the sparsest near-solution. *Communications on Pure and Applied Mathematics*, 59:1207–1223, 2005.

Ethan R. Elenberg, Rajiv Khanna, Alexandros G. Dimakis, and Sahand Negahban. Restricted strong convexity implies weak submodularity. *Annals of Statistics*, 2018. To appear.

Anna Gilbert, S. Muthu Muthukrishnan, and Martin Strauss. Approximation of functions over redundant dictionaries using coherence. In *Proc. 14th ACM-SIAM Symp. on Discrete Algorithms*, pages 243–252, 2003.

Alexander Grubb and J. Andrew Bagnell. Speedboost: Anytime prediction with uniform near-optimality. In *Proc. 15thIntl. Conf. on Artificial Intelligence and Statistics*, pages 458–466, 2012.

Rishabh K. Iyer. *Submodular Optimization and Machine Learning: Theoretical Results, Unifying and Scalable Algorithms, and Applications*. PhD thesis, University of Washington, 2015.

Rishabh K. Iyer and Jeff A. Bilmes. Submodular optimization with submodular cover and submodular knapsack constraints. In *Proc. 27th Advances in Neural Information Processing Systems*, pages 2436–2444, 2013.

Rishabh K. Iyer, Stefanie Jegelka, and Jeff A. Bilmes. Curvature and optimal algorithms for learning and minimizing submodular functions. In *Proc. 27th Advances in Neural Information Processing Systems*, pages 2742–2750, 2013.

Richard A. Johnson and Dean W. Wichern. *Applied Multivariate Statistical Analysis*. Prentice Hall, 2002.

Kwangmoo Koh, Seung-Jean Kim, and Stephen Boyd. l1_ls: Simple Matlab Solver for l1-regularized Least Squares Problems, 2008. http://www.stanford.edu/ boyd/l1_ls.

Andreas Krause and Volkan Cevher. Submodular dictionary selection for sparse representation. In *Proc. 27th Intl. Conf. on Machine Learning*, pages 567–574, 2010.

Andreas Krause and Daniel Golovin. Submodular function maximization. In Lucas Bordeaux, Youssef Hamadi, and Pushmeet Kohli, editors, *Tractability: Practical Approaches to Hard Problems*, pages 71–104. Cambridge University Press, 2014.

Matt J. Kusner, Wenlin Chen, Quan Zhou, Zhixiang Xu, Kilian Q. Weinberger, and Yixin Chen. Feature-cost sensitive learning with submodular trees of classifiers. In *Proc. 28th AAAI Conf. on Artificial Intelligence*, pages 1949–1945, 2014.

Aurelie C. Lozano, Grzegorz Swirszcz, and Naoki Abe. Grouped orthogonal matching pursuit for variable selection and prediction. In *Proc. 23rd Advances in Neural Information Processing Systems*, pages 1150–1158, 2009.

Alan J. Miller. *Subset Selection in Regression.* Chapman and Hall, second edition, 2002.

Balas K. Natarajan. Sparse approximation solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234, 1995.

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

Chao Qian, Yang Yu, and Zhi-Hua Zhou. Subset selection by pareto optimization. In *Proc. 29th Advances in Neural Information Processing Systems*, pages 1774–1782, 2015.

Mahyar Salek, Shahin Shayandeh, and David Kempe. You share, I share: Network effects and economic incentives in P2P file-sharing systems. In *Proc. 6th Workshop on Internet and Network Economics (WINE)*, pages 354–365, 2010.

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

Maxim Sviridenko, Jan Vondrák, and Justin Ward. Optimal approximation for submodular and supermodular optimization with bounded curvature. In *Proc. 26th ACM-SIAM Symp. on Discrete Algorithms*, pages 1134–1148, 2015.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58(1):267–288, 1996.

Joel Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50:2231–2242, 2004.

Joel Tropp. Just relax: Convex programming methods for identifying sparse signals. *IEEE Transactions on Information Theory*, 51:1030–1051, 2006.

Joel Tropp, Anna Gilbert, S. Muthu Muthukrishnan, and Martin Strauss. Improved sparse approximation over quasi-incoherent dictionaries. In *Proc. IEEE-ICIP*, pages 37–40, 2003.

Adrian Vetta. Nash equlibria in competitive societies with applications to facility location, traffic routing and auctions. In *Proc. 43rd IEEE Symp. on Foundations of Computer Science*, pages 416–425, 2002.

Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proc. 40th ACM Symp. on Theory of Computing*, pages 67–74, 2008.

Jan Vondrák. Submodularity and curvature: the optimal algorithm. *RIMS Kokyuroku Bessatsu*, B23:253–266, 2010.

Laurence A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2:385–393, 1982.

Tong Zhang. Adaptive forward-backward greedy algorithm for sparse learning with linear models. In *Proc. 22nd Advances in Neural Information Processing Systems*, pages 1921–1928, 2008.

Tong Zhang. On the consistency of feature selection using greedy least squares regression. *Journal of Machine Learning Research*, 10:555–568, 2009.

Peng Zhao and Bin Yu. On model selection consistency of lasso. *Journal of Machine Learning Research*, 7:2451–2457, 2006.

Shuheng Zhou. Thresholding procedures for high dimensional variable selection and statistical estimation. In *Proc. 23rd Advances in Neural Information Processing Systems*, pages 2304–2312, 2009.

Appendix A. Estimating $\lambda_{\min}(C,k)$

Several of our approximation guarantees are phrased in terms of $\lambda_{\min}(C,k)$. Finding the exact value of $\lambda_{\min}(C,k)$ is **NP**-hard in general; here, we show how to estimate lower and upper bounds. Let $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ be the eigenvalues of C, and $\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_n}$ the corresponding eigenvectors. A first simple bound can be obtained directly from the eigenvalue interlacing theorem: $\lambda_1 \leq \lambda_{\min}(C,k) \leq \lambda_{n-k+1}$.

One case in which good lower bounds on $\lambda_{\min}(C,k)$ can possibly be obtained is when only a small (constant) number of the λ_i are small. The following lemma allows a bound in terms of any λ_j; however, since the running time by the implied algorithm is exponential in j, and the quality of the bound depends on λ_j, it is useful only in the special case when $\lambda_j \gg 0$ for a small constant j.

Lemma 26 *Let V_j be the vector space spanned by the eigenvectors $\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_j}$, and define*

$$\beta_j = \max_{\mathbf{y} \in V_j, \mathbf{x} \in \mathbb{R}^n, \|\mathbf{y}\|_2 = \|\mathbf{x}\|_2 = 1, \|\mathbf{x}\|_0 \leq k} |\mathbf{x} \cdot \mathbf{y}|.$$

Then, $\lambda_{\min}(C,k) \geq \lambda_{j+1} \cdot (1 - \beta_j)$.

Proof. Let $\mathbf{x}' \in \mathbb{R}^n, \|\mathbf{x}'\|_2 = 1, \|\mathbf{x}\|_0 \leq k$ be an eigenvector corresponding to $\lambda_{\min}(C,k)$. Let α_i be the coefficients of the representation of $\mathbf{x}'$ in terms of the $\mathbf{e_i}$: $\mathbf{x}' = \sum_{i=1}^n \alpha_i \mathbf{e_i}$. Thus, $\sum_{i=1}^n \alpha_i^2 = 1$, and we can write

$$\lambda_{\min}(C,k) \;=\; \mathbf{x}'^{\mathsf{T}} C \mathbf{x}' \;=\; \sum_{i=1}^{n} \alpha_i^2 \lambda_i \;\geq\; \lambda_{j+1}(1 - \sum_{i=1}^{j} \alpha_i^2).$$

Since $\sum_{i=1}^j \alpha_i^2$ is the length of the projection of $\mathbf{x}$ onto V_j, we have

$$\sum_{i=1}^{j} \alpha_i^2 \;=\; \max_{\mathbf{y}\in V_j, \|\mathbf{y}\|_2=1} |\mathbf{x}' \cdot \mathbf{y}| \;\leq\; \max_{\mathbf{y}\in V_j, \mathbf{x}\in\mathbb{R}^n, \|\mathbf{x}\|_2=\|\mathbf{y}\|_2=1, \|\mathbf{x}\|_0\leq k} |\mathbf{y}\cdot\mathbf{x}|,$$

completing the proof. ■

Since all the λ_j can be computed easily, the crux in using this bound is finding a good bound on β_j. Next, we show a PTAS (Polynomial-Time Approximation Scheme) for approximating β_j, for any constant j.

Lemma 27 *For every $\epsilon > 0$, there is a $1-\epsilon$ approximation for calculating β_j, running in time $O((\frac{n}{\epsilon})^j)$.*

Proof. Any vector $\mathbf{y} \in V_j$ with $\|\mathbf{y}\|_2 = 1$ can be written as $\mathbf{y} = \sum_{i=1}^j \eta_i \mathbf{e_i}$ with $\eta_i \in [-1,1]$ for all i. The idea of our algorithm is to exhaustively search over all $\mathbf{y}$, as parametrized by their η_i entries. To make the search finite, the entries are discretized to multiples of $\delta = \epsilon \cdot \sqrt{k/(nj)}$. The total number of such vectors to search over is $(2/\delta)^j \leq (n/\epsilon)^j$.

Let $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ attain the maximum in the definition of β_j, and write $\hat{\mathbf{y}} = \sum_{i=1}^j \hat{\eta}_i \mathbf{e_i}$. For each i, let η_i be $\hat{\eta}_i$, rounded to the nearest multiple of δ, and $\mathbf{y} = \sum_{i=1}^j \eta_i \mathbf{e_i}$. Then, $\|\hat{\mathbf{y}} - \mathbf{y}\|_2 \leq \|\delta \sum_{i=1}^j \mathbf{e_j}\|_2 = \delta\sqrt{j}$.

The vector $\mathbf{x}' = \text{argmax}_{\mathbf{x}\in\mathbb{R}^n, \|\mathbf{x}\|_2=1, \|\mathbf{x}\|_0\leq k} |\mathbf{y}\cdot\mathbf{x}|$ is of the following form: Let I be the set of k indices i such that $|y_i|$ is largest, and $\gamma = \sqrt{\sum_{i\in I} y_i^2}$. Then, $x'_i = 0$ for $i \notin I$ and $x'_i = y_i/\gamma$ for $i \in I$. Notice that given $\mathbf{y}$, we can easily find $\mathbf{x}'$, and because $|\hat{\mathbf{x}}\cdot\mathbf{y}| \leq |\mathbf{x}'\cdot\mathbf{y}| \leq |\hat{\mathbf{x}}\cdot\hat{\mathbf{y}}|$, we have

$$\frac{||\hat{\mathbf{x}}\cdot\hat{\mathbf{y}}| - |\mathbf{x}'\cdot\mathbf{y}||}{|\hat{\mathbf{x}}\cdot\hat{\mathbf{y}}|} \;\leq\; \frac{||\hat{\mathbf{x}}\cdot\hat{\mathbf{y}}| - |\hat{\mathbf{x}}\cdot\mathbf{y}||}{|\hat{\mathbf{x}}\cdot\hat{\mathbf{y}}|} \;\leq\; \frac{\|\hat{\mathbf{x}}\|_2\|\hat{\mathbf{y}}-\mathbf{y}\|_2}{|\hat{\mathbf{x}}\cdot\hat{\mathbf{y}}|} \;\leq\; \frac{\delta\sqrt{j}}{|\hat{\mathbf{x}}\cdot\hat{\mathbf{y}}|} \;\leq\; \delta\sqrt{jn/k}.$$

The last inequality follows since the sum of the k largest entries of $\hat{\mathbf{y}}$ is at least $k/\sqrt{n}$, so by setting $x_i = 1/\sqrt{k}$ for each of those coordinates, we can attain at least an inner product of $\sqrt{k/n}$, and the inner product with $\hat{\mathbf{x}}$ cannot be smaller.

The value output by the exhaustive search over all discretized values is at least $|\mathbf{x}'\cdot\mathbf{y}|$, and thus within a factor of $1 - \frac{\delta\sqrt{j}n}{k} = 1-\epsilon$ of the maximum value, attained by $\hat{\mathbf{x}}, \hat{\mathbf{y}}$. ■