On Approximation Guarantees for Greedy Low Rank Optimization

Rajiv Khanna [1] **Ethan R. Elenberg** [1] **Alexandros G. Dimakis** [1] **Joydeep Ghosh** [1] **Sahand Negahban** [2]

Abstract

We provide new approximation guarantees for greedy low rank matrix estimation under standard assumptions of restricted strong convexity and smoothness. Our novel analysis also uncovers previously unknown connections between the low rank estimation and combinatorial optimization, so much so that our bounds are reminiscent of corresponding approximation bounds in submodular maximization. Additionally, we also provide statistical recovery guarantees. Finally, we present empirical comparison of greedy estimation with established baselines on two important real-world problems.

1. Introduction

Low rank matrix optimization stands as a major tool in modern dimensionality reduction and unsupervised learning. The singular value decomposition can be used when the optimization objective is rotationally invariant to the parameters. However, if we wish to optimize over more complex, non-convex objectives we must choose to either rely on convex relaxations (Recht et al., 2010; Negahban & Wainwright, 2011; Rohde & Tsybakov, 2011) or directly optimize over the non-convex space (Park et al., 2016; Jain et al., 2013; Chen & Wainwright, 2015; Lee & Bresler, 2013; Jain et al., 2014).

More concretely, in the low rank matrix optimization problem, we wish to solve

$$\arg\max_{\Theta} \ell(\Theta) \quad \text{s.t. } \text{rank}(\Theta) \leq r. \tag{1}$$

Rather than perform the computationally intractable optimization above researchers have studied convex relaxations of the form

$$\arg\max_{\Theta} \ell(\Theta) - \lambda \|\Theta\|_{\text{nuc}}.$$

Unfortunately, the above optimization can be computationally taxing. General purpose solvers for the above optimization problem that rely on semidefinite programming (SDP) require $\Theta(n^3d^3)$ computation, which is prohibitive. Gradient descent techniques require $\Theta(\epsilon^{-1/2}(n^3 + d^3))$ computational cost for an epsilon accurate solution. This improvement is sizable in comparison to SDP solvers. Unfortunately, it is still prohibitive for large scale matrix estimation.

An alternate vein of research has focused on directly optimizing the non-convex problem (1). To that end, authors have seen recent theoretical success in studying the convergence properties of

$$\arg\max_{\mathbf{U}\in\mathbb{R}^{n\times r},\mathbf{V}\in\mathbb{R}^{d\times r}} \ell(\mathbf{U}\mathbf{V}^T).$$

Solving the problem above automatically forces the solution to be low rank, and recent results have shown promising behavior.

Another approach is to optimize (1) *incrementally* via rank one updates to the current estimate (Shalev-Shwartz et al., 2011; Wang et al., 2015). This approach has also been studied in more general contexts such as boosting (Buhlmann & Yu, 2009), coordinate descent (Jaggi, 2013; Jaggi & Sulovský, 2010), and incremental atomic norm optimization (Gribonval & Vandergheynst, 2006; Barron et al., 2008; Khanna et al., 2016; Rao et al., 2015; Dudik et al., 2012; Locatello et al., 2017).

1.1. Set Function Optimization and Coordinate Descent

In this paper, we interpret low rank matrix estimation as a set optimization problem over an infinite set of atoms. Specifically, we wish to optimize

$$\arg\max_{\{\mathbf{X}_1,\ldots\mathbf{X}_k\}\in\mathcal{A}} \ell\left(\sum_{i=1}^{k}\alpha_i\mathbf{X}_i\right),$$

where the set of atoms $\mathcal{A}$ is the set of all rank one matrices with unit operator norm. This settings is analogous to that taken in the results studying atomic norm optimization, coordinate descent via the total variation norm, and Frank-Wolfe style algorithms for atomic optimization. This formulation allows us to connect the problem of low rank matrix estimation to that of submodular set function optimization, which

[1]UT Austin [2]Yale University. Correspondence to: Rajiv Khanna <rajivak@utexas.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

we discuss in the sequel. Before proceeding we discuss related work and an informal statement of our main result.

1.2. Informal Result and Related Work

Our result demonstrates an exponential decrease in the amount of error incurred by greedily adding rank one matrices to the low rank matrix approximation.

Theorem 1 (Approximation Guarantee, Informal)**.** *If we let Θ_k be our estimate of the rank r matrix Θ^* at iteration k, then for some universal constant c related to the restricted condition number of the problem we have*

$$\ell(\Theta_k) - \ell(\mathbf{0}) \geq (1 - \exp(-ck/r))(\ell(\Theta^*) - \ell(\mathbf{0})).$$

Note that after k iterations the matrix Θ_k will be at most rank k.

Related work: There has been a wide array of studies looking at the computational and statistical benefits of rank one updates to estimating a low rank matrix. At its most basic, the singular value decomposition will keep adding rank one approximations through deflation steps. This work can be generally segmented into two sets of results – the ones that present sublinear rates of convergence and those that obtain linear rates. Interestingly, parallel lines of work have also demonstrated similar convergence bounds for more general atomic or dictionary element approximations (Buhlmann & Yu, 2009; Gribonval & Vandergheynst, 2006; Barron et al., 2008; Khanna et al., 2016). For space constraints, we will summarize these results into two categories rather than explicitly state the results for each individual paper.

If we define the atomic norm of a matrix $\mathbf{M} \in \mathbb{R}^{n \times d}$ written as $\|\mathbf{M}\|_{\text{nuc}}$ to be the sum of the singular values of that matrix, then the bounds establishing sublinear convergence behave as

$$\ell(\Theta^*) - \ell(\Theta_k) \leq \frac{\|\Theta^*\|_{\text{nuc}}^2}{k},$$

where we take Θ^* to be the best rank r solution. What we then see is convergence towards the optimal bound. However, we expect our statistical error to behave as $r(n+d)/n$ where n is the number of samples that we have received from our statistical model and Θ^* is rank r (Negahban & Wainwright, 2011; Rohde & Tsybakov, 2011). We can take $\|\Theta^*\|_{\text{nuc}} \approx r$, which would then imply that we would need k to behave as $n/(n+d)$. However, that would then imply that the rank of our matrix should grow linearly in the number of observations in order to achieve the same statistical error bounds. The above error bounds are "fast". If we consider a model that yields slow error bounds, then we expect the error to behave like $\|\Theta^*\|_{\text{nuc}}\sqrt{\frac{n+d}{n}}$. In that case, we can take $k \geq \|\Theta^*\|_{\text{nuc}}\sqrt{\frac{n}{n+d}}$, which looks better, but still requires significant growth in k as a function of n. To overcome the above points, some authors have aimed to study similar greedy algorithms that then enjoy exponential rates of convergence as we show in our paper. These results share the most similarities with our own and behave as

$$\ell(\Theta_k) \geq (1 - \gamma^k)\ell(\Theta^*).$$

This result decays exponentially. However, when one looks at the behavior of γ it will typically act as $\exp(-1/\min(n,d))$, for an $n \times d$ matrix. As a result, we would need to choose k of the order of the dimensionality of the problem in order to begin to see gains. In contrast, for our result listed above, if we seek to only compare to the best rank r solution, then the gamma we find is $\gamma = \exp(-1/r)$. Of course, if we wish to find a solution with full rank, then the bounds we stated above match the existing bounds.

In order to establish our results we rely on a notion introduced in the statistical community called restricted strong convexity. This assumption has connections to ideas such as the restricted isometry property, restricted eigenvalue condition, and incoherence (Negahban & Wainwright, 2012). In the work by Shalev-Shwartz, Gonen, and Shamir (2011) a form of strong convexity condition is imposed over matrices. Under that setting, the authors demonstrate that

$$\ell(\Theta_k) \geq \ell(\Theta^*) - \frac{\ell(\mathbf{0})r}{k},$$

where r is the rank of Θ^*. In contrast, our bound behaves as

$$\ell(\Theta_k) \geq \ell(\Theta^*) - (\ell(\Theta^*) - \ell(\mathbf{0}))\exp(-k/r).$$

Our contributions: We improve upon the linear rates of convergence for low rank approximation using rank one updates by connecting the coordinate descent problem to that of submodular optimization. We present this result in the sequel along with the algorithmic consequences. We demonstrate the good performance of these rank one updates in the experimental section.

2. Background

We begin by fixing some notation. We represent sets using sans script fonts *e.g.* A, B. Vectors are represented using lower case bold letters *e.g.* $\mathbf{x}, \mathbf{y}$, and matrices are represented using upper case bold letters *e.g.* $\mathbf{X}, \mathbf{Y}$. Non-bold face letters are used for scalars *e.g.* j, M, r and function names *e.g.* $f(\cdot)$. The transpose of a vector or a matrix is represented by $\top$ *e.g.* $\mathbf{X}^\top$. Define $[p] := \{1, 2, \ldots, p\}$. For singleton sets, we write $f(j) := f(\{j\})$. Size of a set S is denoted by $|\mathsf{S}|$. $\langle \cdot, \cdot \rangle$ is used for matrix inner product.

Our goal is to analyze greedy algorithms for low rank estimation. Consider the classic greedy algorithm that picks up

the next element *myopically i.e.* given the solution set built so far, the algorithm picks the next element as the one which maximizes the gain obtained by adding the said element into the solution set. Approximation guarantees for the greedy algorithm readily imply for the class of functions defined as follows.

Definition 1. *A set function $f(\cdot) : [p] \to \mathbb{R}$ is submodular if for all $\mathsf{A}, \mathsf{B} \subseteq [p]$,*

$$f(\mathsf{A}) + f(\mathsf{B}) \geq f(\mathsf{A} \cup \mathsf{B}) + f(\mathsf{A} \cap \mathsf{B}).$$

Submodular set functions are well studied and have many desirable properties that allow for efficient minimization and maximization with approximation guarantees. Our low rank estimation problem also falls under the purview of another class of functions called *monotone* functions. A function is called monotone if and only if $f(\mathsf{A}) \leq f(\mathsf{B})$ for all $\mathsf{A} \subseteq \mathsf{B}$. For the specific case of maximizing monotone submodular set functions, it is known that the greedy algorithm run for k iterations is guaranteed to return a solution that is within $(1 - 1/e)$ of the optimum set of size k (Nemhauser et al., 1978). Without further assumptions or knowledge of the function, no other polynomial time algorithm can provide a better approximation guarantee unless P=NP (Feige, 1998).

More recently, the aforementioned greedy approximation guarantee has been extended to a larger class of functions called *weakly* submodular functions (Elenberg et al., 2016; Khanna et al., 2017). Central to the notion of weak submodularity is a quantity called the submodularity ratio.

Definition 2 (Submodularity Ratio (Das & Kempe, 2011))**.** *Let $\mathsf{S}, \mathsf{L} \subset [p]$ be two disjoint sets, and $f(\cdot) : [p] \to \mathbb{R}$. The submodularity ratio of L with respect to S is given by*

$$\gamma_{\mathsf{L},\mathsf{S}} := \frac{\sum_{j \in \mathsf{S}} [f(\mathsf{L} \cup \{j\}) - f(\mathsf{L})]}{f(\mathsf{L} \cup \mathsf{S}) - f(\mathsf{L})}. \tag{2}$$

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

$$\gamma_{\mathsf{U},k} := \min_{\substack{\mathsf{L},\mathsf{S}:\mathsf{L} \cap \mathsf{S} = \emptyset, \\ \mathsf{L} \subseteq \mathsf{U}, |\mathsf{S}| \leq k}} \gamma_{\mathsf{L},\mathsf{S}}. \tag{3}$$

It is easy to show that $f(\cdot)$ is submodular if and only if $\gamma_{\mathsf{L},\mathsf{S}} \geq 1$ for all sets L and S. However, an approximation guarantee is obtained when $0 < \gamma_{\mathsf{L},\mathsf{S}} \ \ \forall \ \mathsf{L}, \mathsf{S}$ (Das & Kempe, 2011; Elenberg et al., 2016). The subset of monotone functions which have $\gamma_{\mathsf{L},\mathsf{S}} > 0 \ \ \forall \ \mathsf{L}, \mathsf{S}$ are called weakly submodular functions in the sense that even though the function is not submodular, it still provides a provable bound for greedy selections.

Also vital to our analysis is the notion of restricted strong concavity and smoothness (Negahban et al., 2012; Loh & Wainwright, 2015).

Definition 3 (Low Rank Restricted Strong Concavity (RSC), Restricted Smoothness (RSM))**.** *A function $\ell : \mathbb{R}^{n \times d} \to \mathbb{R}$ is said to be restricted strong concave with parameter m_Ω and restricted smooth with parameter M_Ω if for all $\mathbf{X}, \mathbf{Y} \in \Omega \subset \mathbb{R}^{n \times d}$,*

$$-\frac{m_\Omega}{2}\|\mathbf{Y} - \mathbf{X}\|_F^2 \geq \ell(\mathbf{Y}) - \ell(\mathbf{X}) - \langle \nabla \ell(\mathbf{X}), \mathbf{Y} - \mathbf{X} \rangle \geq -\frac{M_\Omega}{2}\|\mathbf{Y} - \mathbf{X}\|_F^2.$$

Remark 1. *If a function $\ell(\cdot)$ has restricted strong concavity parameter m, then its negative $-\ell(\cdot)$ has restricted strong convexity parameter m. We choose to use the nomenclature of concavity for ease of exposition in terms of relationship to submodular maximization. Further, note that we define RSC/RSM conditions on the space of matrices rather than vectors, on a domain Ω constrained by rank rather than sparsity. It is straightforward to see that if $\Omega' \subseteq \Omega$, then $M_{\Omega'} \leq M_\Omega$ and $m_{\Omega'} \geq m_\Omega$.*

3. Setup

In this section, we delineate our setup of low rank estimation. In order to connect to the weak submodular maximization framework more easily, we operate in the setting of maximization of a concave matrix variate function under a low rank constraint. This is equivalent to minimizing a convex matrix variate function under the low rank constraint as considered by Shalev-Shwartz et al. (2011) or under nuclear norm constraint or regularization as considered by Jaggi & Sulovský (2010). The goal is to maximize a function $\ell : \mathbb{R}^{n \times d} \to \mathbb{R}$:

$$\max_{\text{rank}(\mathbf{X}) \leq r} \ell(\mathbf{X}). \tag{4}$$

Instead of using a convex relaxation of (4), our approach is to enforce the rank constraint directly by adding rank 1 matrices greedily until $\mathbf{X}$ is of rank k. The rank 1 matrices to be added are obtained as outer product of vectors from the given vector sets $\mathcal{U}$ and $\mathcal{V}$. While our results hold for general vector sets $\mathcal{U}, \mathcal{V}$ assuming an oracle access to subroutines `GreedySel` and `OMPSel` (to be detailed later), for the rest of the paper we focus on the case of norm 1 balls $\mathcal{U} := \{\mathbf{x} \in \mathbb{R}^n \, s.t. \, \|\mathbf{x}\|_2 = 1\}$ and $\mathcal{V} := \{\mathbf{x} \in \mathbb{R}^d \, s.t. \, \|\mathbf{x}\|_2 = 1\}$.

The problem (4) can be interpreted in the context of sparsity assuming $\mathcal{U}$ and $\mathcal{V}$ are enumerable. For example, by the SVD theorem, it is known that we can rewrite $\mathbf{X}$ as $\sum_{i=1}^k \alpha_i \mathbf{u}_i \mathbf{v}_i^\top$, where $\forall i$, $\mathbf{u}_i \in \mathcal{U}$ and $\mathbf{v}_i \in \mathcal{V}$. By enumerating $\mathcal{U}$ and $\mathcal{V}$ under a finite precision representation of real values, one can rethink of the optimization (4) as finding a sparse solution for the infinite dimensional vector $\boldsymbol{\alpha}$ (Shalev-Shwartz et al., 2011; Dudik et al., 2012). We can also optimize over support sets, similar to the classical setting of support selection for sparse vectors. For a *specified* support set L consisting of vectors from $\mathcal{U}$ and $\mathcal{V}$, let

$\mathbf{U}_{\mathsf{L}}$ and $\mathbf{V}_{\mathsf{L}}$ be the matrices formed by stacking the chosen elements of $\mathcal{U}$ and $\mathcal{V}$ respectively. We define the following set function to maximize $\ell(\cdot)$ given L.

$$f(\mathsf{L}) = \max_{\mathbf{H}\in\mathbb{R}^{|\mathsf{L}|\times|\mathsf{L}|}} \ell(\mathbf{U}_{\mathsf{L}}^\top \mathbf{H}\mathbf{V}_{\mathsf{L}}) - \ell(\mathbf{0}). \tag{5}$$

We will denote the optimizing matrix for a support set L as $\mathbf{B}^{(\mathsf{L})}$. In other words, letting $\hat{\mathbf{H}}_{\mathsf{L}}$ be the argmax obtained in (5), then $\mathbf{B}^{(\mathsf{L})} := \mathbf{U}_{\mathsf{L}}^\top \hat{\mathbf{H}}_{\mathsf{L}} \mathbf{V}_{\mathsf{L}}$. Thus, the low rank matrix estimation problem (4) can be reinterpreted as the following equivalent combinatorial optimization problem:

$$\max_{|\mathsf{S}|\le k} f(\mathsf{S}). \tag{6}$$

3.1. Algorithms

Our greedy algorithm, illustrated in Algorithm 1, builds the support set incrementally – adding rank 1 matrices one at a time such that at iteration i for $1 \le i \le k$ the size of the chosen support set (and hence rank of the current iterate) is i. We assume access to a subroutine `GreedySel` for the greedy selection (Step 4). This subroutine solves an inner optimization problem by calling a subroutine `GreedySel` which returns an atom s from the candidate support set that ensures

$$f(\mathsf{S}^G_{i-1}\cup\{s\}) - f(\mathsf{S}^G_{i-1}) \ge \tau\left(f(\mathsf{S}^G_{i-1}\cup\{s^\star\}) - f(\mathsf{S}^G_{i-1})\right),$$

where

$$s^\star \leftarrow \operatorname*{arg\,max}_{a\in(\mathcal{U}\times\mathcal{V})\perp\mathsf{S}^G_{i-1}} f(\mathsf{S}^G_{i-1}\cup\{a\}) - f(\mathsf{S}^G_{i-1}).$$

In words, the subroutine `GreedySel` ensures that the gain in $f(\cdot)$ obtained by using the selected atom is within $\tau \in (0,1]$ multiplicative approximation to the atom with the best possible gain in $f(\cdot)$. The hyperparameter τ governs a tradeoff allowing a compromise in myopic gain for a possibly quicker selection.

The greedy selection requires fitting and scoring every candidate support, which is often prohibitively expensive. An alternative is to choose the next atom by using the linear maximization oracle used by Frank-Wolfe (Jaggi, 2013) or Matching Pursuit algorithms (Gribonval & Vandergheynst, 2006; Locatello et al., 2017). This step replaces Step 4 of Algorithm 1 as illustrated in Algorithm 2. Let $\mathsf{L} = \mathsf{S}^O_{i-1}$ be the set constructed by the algorithm at iteration $(i-1)$. The linear oracle `OMPSel` returns an atom s for iteration i ensuring

$$\langle \nabla\ell(\mathbf{B}^{(\mathsf{L})}), \mathbf{u}_s\mathbf{v}_s^\top\rangle \ge \tau \max_{(\mathbf{u},\mathbf{v})\in(\mathcal{U}\times\mathcal{V})\perp\mathsf{S}^O_{i-1}} \langle \nabla\ell(\mathbf{B}^{(\mathsf{L})}), \mathbf{u}\mathbf{v}^\top\rangle.$$

The linear problem `OMPSel` can be considerably faster that `GreedySel`. `OMPSel` reduces to finding the left and right singular vectors of $\nabla\ell(\mathbf{B}^{(\mathsf{L})})$ corresponding to its largest singular value. If t is the number of non-zero entries in $\nabla\ell(\mathbf{B}^{(\mathsf{L})})$, then this takes $O(\frac{t}{1-\tau}(\log n + \log d))$ time.

We note that Algorithm 2 is the same as considered by Shalev-Shwartz et al. (2011) as GECO (Greedy Efficient Component Optimization). However, as we shall see, our analysis provides stronger bounds than their Theorem 2.

Algorithm 1 GREEDY($\mathcal{U}$, $\mathcal{V}$, k, τ)

1: **Input:** vector sets $\mathcal{U}$, $\mathcal{V}$, sparsity parameter k, subroutine hyperparameter τ
2: $\mathsf{S}^G_0 \leftarrow \emptyset$
3: **for** $i = 1 \dots k$ **do**
4: $\quad s \leftarrow$ `GreedySel`(τ)
5: $\quad \mathsf{S}^G_i \leftarrow \mathsf{S}^G_{i-1} \cup \{s\}$
6: **end for**
7: **return** S^G_k, $\mathbf{B}^{(\mathsf{S}^G_k)}$, $f(\mathsf{S}^G_k)$.

Algorithm 2 GECO($\mathcal{U}$, $\mathcal{V}$, k, τ)

same as Algorithm 1 except
4: $s \leftarrow$ `OMPSel`(τ)

Remark 2. *We note that Step 5 of Algorithms 1 and 2 requires solving the RHS of (5) which is a matrix variate problem of size i^2 at iteration i. This refitting is equivalent to the "fully-corrective" versions of Frank-Wolfe/Matching Pursuit algorithms (Locatello et al., 2017; Lacoste-Julien & Jaggi, 2015) which, intuitively speaking, extract out all the information w.r.t $\ell(\cdot)$ from the chosen set of atoms, thereby ensuring that the next rank 1 atom chosen has row and column space orthogonal to the previously chosen atoms. Thus the constrained maximization on the orthogonal complement of S^O_i in subroutine `OMPSel` (S^G_i in `GreedySel`) need not be explicitly enforced, but is still shown for clarity.*

4. Analysis

In this section, we prove that low rank matrix optimization over the rank one atoms satisfies weak submodularity. We explicitly delineate some notation and assumptions. With slight abuse of notation, we assume $\ell(\cdot)$ is m_i-strongly concave and M_i-smooth over pairs of matrices of rank i. For $i \le j$, note that $m_i \ge m_j$ and $M_i \le M_j$. Additionally, let $\tilde{\Omega} := \{(\mathbf{X},\mathbf{Y}) : \text{rank}(\mathbf{X}-\mathbf{Y}) \le 1\}$, and assume $\ell(\cdot)$ is $\tilde{M}_1$-smooth over $\tilde{\Omega}$. It is easy to see $\tilde{M}_1 \le M_1$.

First we prove that if the low rank RSC holds (Definition 3), then the submodularity ratio (Definition 2) is lower-bounded by the inverse condition number.

Theorem 2. *Let L be a set of k rank 1 atoms and S be a set of r rank 1 atoms where we sequentially orthogonalize the atoms against L. If $\ell(\cdot)$ is m_i-strongly con-*

cave over matrices of rank i, and $\tilde{M}_1$-smooth over the set $\tilde{\Omega} := \{(\mathbf{X}, \mathbf{Y}) : \text{rank}(\mathbf{X} - \mathbf{Y}) = 1\}$, then

$$\gamma_{\mathsf{L},r} := \frac{\sum_{a \in S}[f(\mathsf{L} \cup \{a\}) - f(\mathsf{L})]}{f(\mathsf{L} \cup \mathsf{S}) - f(\mathsf{L})} \geq \frac{m_{r+k}}{\tilde{M}_1}.$$

The proof of Theorem 2 is structured around individually obtaining a lower bound for the numerator and an upper bound for the denominator of the submodularity ratio by exploiting the concavity and convexity conditions.

4.1. Greedy Improvement

Bounding the submodularity ratio is crucial to obtaining approximation guarantees for Algorithm 1.

Theorem 3. *Let $\mathsf{S} := \mathsf{S}_k^G$ be the greedy solution set obtained by running Algorithm 1 for k iterations, and let $\mathsf{S}^\star$ be an optimal support set of size r. Let $\ell(\cdot)$ be m_i strongly concave on the set of matrices with rank less than or equal to i, and $\tilde{M}_1$ smooth on the set of matrices in the set $\tilde{\Omega}$. Then,*

$$f(\mathsf{S}) \geq \left(1 - \frac{1}{e^{c_1}}\right) f(\mathsf{S}^\star) \geq \left(1 - \frac{1}{e^{c_2}}\right) f(\mathsf{S}^\star),$$

where $c_1 = \tau \gamma_{\mathsf{S},r} \frac{k}{r}$ and $c_2 = \tau \frac{m_{r+k}}{\tilde{M}_1} \frac{k}{r}$.

The proof technique for the first inequality of Theorem 3 relies on lower bounding the progress made in each iteration of Algorithm 1. Intuitively, it exploits weak submodularity to make sure that each iteration makes *enough* progress, and then applies an induction argument for r iterations. We also emphasize that the bounds in Theorem 3 are for *normalized* set function $f(\cdot)$ (which means $f(\emptyset) = 0$). A more detailed proof is presented in the appendix.

The bounds obtained in Theorem 3 are similar to the one obtained in submodular maximization of monotone normalized functions (Nemhauser et al., 1978). In fact, our result can be re-interpreted as an extension to previous results. The greedy algorithm for submodular maximization assumes finite ground sets. We extend this for infinite ground sets. We can do this (for matrices) as long as we have an implementation of the oracle `GreedySel`. Once the choice is made by the oracle, standard analysis holds.

Remark 3. *Theorem 3 provides the approximation guarantees for running the greedy selection algorithm up to k iterations to obtain a rank k matrix iterate* vis-a-vis *the best rank r approximation. For $r = k$, and $\tau = 1$, we get an approximation bound $(1 - e^{-m/M})$ which is reminiscent of the greedy bound of $(1 - 1/e)$ under the framework of submodularity. Note that our analysis can not be used to establish classical submodularity. However, establishing weak submodularity that lower bounds γ is sufficient to provide slightly weaker than classical submodularity guarantees.*

Remark 4. *Theorem 3 implies that to obtain $(1-\epsilon)$ approximation guarantee in the worst case, running Algorithm 1 for $k = \frac{rM}{m\tau} \log \frac{1}{\epsilon}) = O(r \log 1/\epsilon)$ iterations suffices. This is useful when the application allows a tradeoff: compromising on the low rank constraint a little to achieve tighter approximation guarantees.*

Remark 5. *Das & Kempe (2011) considered the special case of greedily maximizing R^2 statistic for linear regression, which corresponds to classical sparsity in vectors. They also obtain a bound of $(1 - 1/e^\gamma)$, where γ is the submodularity ratio for their respective setup. This was generalized by Elenberg et al. (2016) to general concave functions under sparsity constraints. Our analysis is for the low rank constraint, as opposed to sparsity in vectors that was considered by them.*

4.2. GECO Improvement

In this section, we obtain approximation guarantees for Algorithm 2. The greedy search over the infinitely many candidate atoms is infeasible, especially when $\tau = 1$. Thus while Algorithm 1 establishes interesting theoretical connections with submodularity, it is not practical in general. To obtain a tractable and practically useful algorithm, the greedy search is replaced by a Frank Wolfe or Matching Pursuit style linear optimization which can be easily implemented as finding the top singular vectors of the gradient at iteration i. In this section, we show that despite the speedup, we lose very little in terms of approximation guarantees. In fact, if the approximation factor τ in `OMPSel()` is 1, we get the same bounds as those obtained for the greedy algorithm.

Theorem 4. *Let $\mathsf{S} := \mathsf{S}_k^O$ be the greedy solution set obtained using Algorithm 2 for k iterations, and let $\mathsf{S}^\star$ be the optimum size r support set. Let $\ell(\cdot)$ be m_{r+k} strongly concave on the set of matrices with rank less than or equal to $(r + k)$, and $\tilde{M}_1$ smooth on the set of matrices with rank in the set $\tilde{\Omega}$. Then,*

$$f(\mathsf{S}) \geq \left(1 - \frac{1}{e^{c_3}}\right) f(\mathsf{S}^\star),$$

where $c_3 = \tau^2 \frac{m_{r+k}}{\tilde{M}_1} \frac{k}{r}$.

The proof of Theorem 4 follows along the lines of Theorem 3. The central idea is similar – to exploit the RSC conditions to make sure that each iteration makes *sufficient* progress, and then provide an induction argument for r iterations. Unlike the greedy algorithm, however, using the submodularity ratio is no longer required. Note that the bound obtained in Theorem 4 is similar to Theorem 3, except the exponent on the approximation factor τ.

Remark 6. *Our proof technique for Theorem 4 can be applied for classical sparsity to improve the bounds obtained by Elenberg et al. (2016) for OMP for support selection*

under RSC, and by Das & Kempe (2011) for R^2 statistic. If $\tau = 1, r = k$, their bounds involve terms of the form $O(m^2/M^2)$ in the exponent, as opposed to our bounds which only has m/M in the exponent.

Remark 7. *Similar to the greedy algorithm, to achieve a tighter approximation to best rank k solution, one can relax the low rank constraint a little by running the algorithm for $r > k$ greedy iterations. The result obtained by our Theorem 4 can be compared to the bound obtained by (Shalev-Shwartz et al., 2011) [Theorem 2] for the same algorithm. For an ϵ multiplicative approximation, Theorem 4 implies we need $r/k = O(\log 1/\epsilon)$. On the other hand, Shalev-Shwartz et al. (2011) obtain an additive approximation bound with $r/k = O(1/\varepsilon)$, which is an exponential improvement.*

5. Recovery Guarantees

While understanding approximation guarantees are useful, providing parameter recovery bounds can further help us understand the practical utility of greedy algorithms. In this section, we present a general theorem that provides us with recovery bounds of the true underlying low rank structure.

Theorem 5. *Suppose that an algorithm achieves the approximation guarantee:*

$$f(\mathsf{S}_k) \geq C_{r,k} f(\mathsf{S}_r^\star),$$

where S_k is the set of size k at iteration k of the algorithm, $\mathsf{S}_r^\star$ be the optimal solution for r-cardinality constrained maximization of $f(\cdot)$, and $C_{r,k}$ be the corresponding approximation ratio guaranteed by the algorithm. Recall that we represent by $\mathbf{U}_\mathsf{S}, \mathbf{V}_\mathsf{S}$ the matrices formed by stacking the vectors represented by the support set S chosen from $\mathcal{U}, \mathcal{V}$ respectively, s.t. $|\mathsf{S}| = r$. Then under m_{k+r} RSC, with $\mathbf{B}_\mathsf{r} = \mathbf{U}_\mathsf{S}^\top \mathbf{H} \mathbf{V}_\mathsf{S}$ for any $\mathbf{H} \in \mathbb{R}^{r \times r}$, we have

$$\|\mathbf{B}^{(\mathsf{S}_k)} - \mathbf{B}_\mathsf{r}\|_F^2 \leq 4(k+r)\frac{\|\nabla \ell(\mathbf{B}_\mathsf{r})\|_2^2}{m_{k+r}^2} + \frac{4(1-C_{r,k})}{m_{k+r}}[\ell(\mathbf{B}_\mathsf{r}) - \ell(\mathbf{0})]$$

Theorem 5 can be applied for $\mathbf{B}_\mathsf{r} = \mathbf{B}^{(\mathsf{S}_r^\star)}$, which is the argmax for maximizing $\ell(\cdot)$ under the low rank constraint. It is general in the sense that it can be applied for getting recovery bounds from approximation guarantees for any algorithm, and hence is applicable for both Algorithms 1 and 2.

Statistical recovery guarantees can be obtained from Theorem 5 for specific choice of $\ell(\cdot)$ and statistical model. Consider the case of low rank matrix estimation from noisy linear measurements. Let $\mathbf{X}_i \in \mathbb{R}^{m_1 \times m_2}$ for $i \in [n]$ be generated so that each entry of $\mathbf{X}_i$ is $\mathcal{N}(0,1)$. We observe $\mathbf{y}_i = \langle \mathbf{X}_i, \Theta^\star \rangle + \varepsilon$, where $\Theta^\star$ is low rank, and say $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. Let $N = m_1 m_2$, and let $\varphi(\Theta) : \mathbb{R}^{m_1 \times m_2} \to \mathbb{R}^n$ be the linear operator so that $[\varphi(\Theta)]_i = \langle \mathbf{X}_i, \Theta \rangle$. Our corresponding function is now $\ell(\Theta) = -\frac{1}{n}\|\mathbf{y} - \varphi(\Theta)\|_2^2$. For this function, using arguments by Negahban et al. (2012), we know $\|\nabla \ell(\mathbf{B}^{\mathsf{S}_r^\star})\|_2^2 \leq \frac{\log N}{n}$ and $\ell(\mathbf{B}^{\mathsf{S}_r^\star}) - \ell(\mathbf{0}) \leq (r+1)$ with high probability. It is also straightforward to apply their results to bound $m_{k+r} \geq \left(\frac{1}{32} - \frac{162(k+r)\log N}{n}\right)$, and $M_1 \leq 1$, which gives explicit bounds as per Theorem 5 for Algorithms 1, 2 for the considered function and the design matrix.

6. Experiments

In this section, we empirically evaluate the proposed algorithms.

6.1. Clustering under Stochastic Block Model

First, we test empirically the performance of GECO (Algorithm 2 with $\tau = 1$) for a clustering task. We are provided with a graph with nodes and the respective edges between the nodes. The observed graph is assumed to have been noisily generated from a true underlying clustering. The goal is to recover the underlying clustering structure from the noisy graph provided to us. Our greedy framework is applicable because the adjacency matrix of the true clustering is low rank. We compare performance of Algorithm 2 on simulated data against standard baselines of spectral clustering which are commonly used for this task. We begin by describing a generative model for creating edges between nodes given the ground truth.

The Stochastic Block Model is a model to generate random graphs. It takes its input the set of n nodes, and a partition of $[n]$ which form a set of disjoint clusters, and returns the graph with nodes and the generated edges. The model has two additional parameters, the generative probabilities (p, q). A pair of nodes within the same cluster have an edge between them with probability p, while a pair of nodes belonging to different clusters have an edge between them with probability q. For simplicity we assume $q = (1-p)$. The model then iterates over each pair of nodes. For each such pair that belongs to same cluster, it samples an edge as Bernoulli(p), otherwise as Bernoulli($1-p$). This provides us with a $\{0,1\}$ adjacency matrix.

We compare against two versions of spectral clustering, which is a standard technique applied to find communities in a graph. The method takes as input the $n \times n$ adjacency matrix $\mathbf{A}$, which is a $\{0,1\}$ matrix with an entry $\mathbf{A}_{ij} = 1$ if there is an edge between node i and j, and is 0 otherwise. From the adjacency matrix, the graph Laplacian $\mathbf{L}$ is constructed. The Laplacian may be unnormalized, in which case it is simply $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D}$ is the diagonal matrix of degrees of nodes. A normalized Laplacian is

computed as $\mathbf{L}_{\text{norm}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$. After calculating the Laplacian, the algorithm solves for bottom k eigenvectors of the Laplacian, and then apply k-means clustering on the rows of the thus obtained eigenvector matrix. We refer to the works of Shi & Malik (2000); Ng et al. (2001) for the specific details of clustering algorithms using unnormalized and normalized graph Laplacian respectively.

We use our greedy algorithm to cluster the graph by optimizing a logistic PCA objective function, which is a special case of the exponential family PCA (Collins et al., 2001). For a given matrix $\mathbf{X}$, each entry $\mathbf{X}_{ij}$ is assumed to be independently drawn with likelihood proportional to $\exp \langle \Theta_{ij}, \mathbf{X}_{ij} \rangle - G(\Theta_{ij})$, where Θ is the true underlying parameter, and $G(\cdot)$ is the partition function corresponding to a generalized linear model (GLM). It is easy to see we can apply our framework of greedy selection by defining $\ell(\cdot)$ as the log-likelihood:

$$\ell(\Theta) = \langle \Theta, \mathbf{X} \rangle - \sum_{i,j} \log G(\Theta_{ij}),$$

where Θ is the true parameter matrix of p and q that generates a realization of $\mathbf{A}$. Since the true Θ is low rank, we get the low rank constrained optimization problem:

$$\max_{\text{rank}(\Theta) \leq k} \ell(\Theta),$$

where k is a hyperparameter suggesting the true number of clusters. Note that lack of knowledge of true value of k is not more restrictive than spectral clustering algorithms which typically also require the true value of k. Having cast the clustering problem in the same form as (4), we can apply our greedy selection algorithm as opposed to the more costly alternating minimizing algorithms suggested by Collins et al. (2001).

We generate the data as follows. For $n = 100$ nodes, and fixed number of cluster $k = 5$, we vary the within cluster edge generation probability p from 0.55 to 0.95 in increments of 0.05, and use the Stochastic Block model to generate a noisy graph with each p. Note that smaller p implies that the sampled graph will be more noisy and likely to be more different than the underlying clustering.

We compare against the spectral clustering algorithm using unnormalized Laplacian of Shi & Malik (2000) which we label "Spectral_unnorm$\{k\}$" for $k = \{3, 5, 10\}$, and the spectral clustering algorithm using normalized Laplacian of Ng et al. (2001) which we label "Spectral_norm$\{k\}$" for $k = \{3, 5, 10\}$. We use Algorithm 2 which we label "Greedy$\{k\}$" for $k = \{3, 5, 10\}$. For each of these models, the referred k is the supplied hyperparameter. We report the least squares error of the output from each model to the true underlying Θ (generalization error), and to the instantiation used for training $\mathbf{X}$ (reconstruction error).

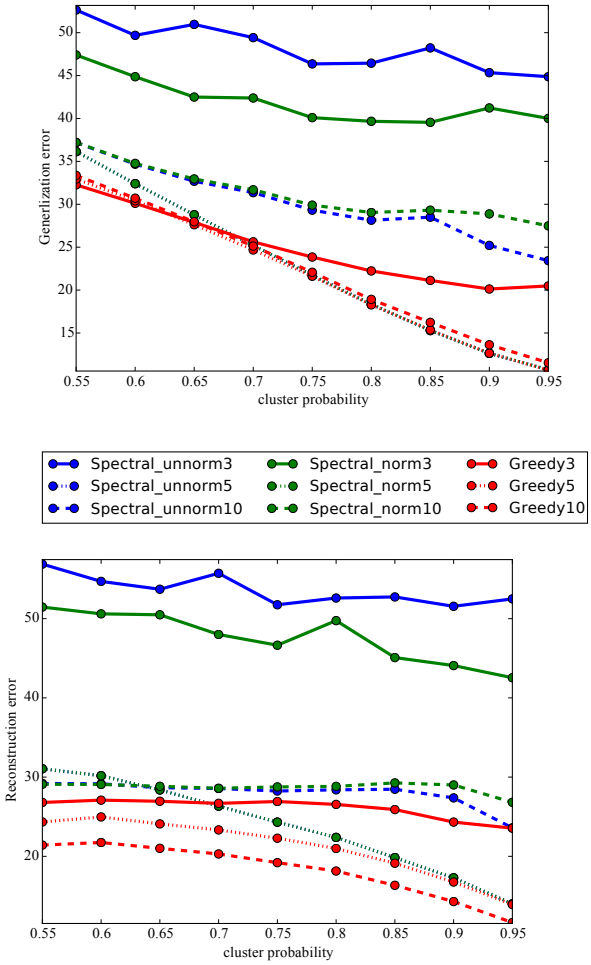

Figure 1. Greedy Logistic PCA vs spectral clustering baselines averaged over 10 runs. *Top*: Robust performance of greedy logistic PCA for generalizing over varying values of k across different values of p, spectral clustering algorithms are more sensitive to knowing true value of k *Bottom*: Strong performance of greedy logisitic PCA even with small value of $k = 3$ for reconstructing the given cluster matrix.

Figure 1 shows that the greedy logistic PCA performs well in not only recreating the given noisy matrix (reconstruction) but also captures the true low rank structure better (generalization). Further, note that providing the true hyperparameter k is vital for spectral clustering algorithms, while on the other hand greedy is less sensitive to k. This is very useful in practice as k is typically not known. Spectral clustering algorithms typically select k by computing an SVD and rerunning k-means for different values of k. In addition to being more robust, our greedy algorithm does not need to be rerun for different values of k – it produces solutions incrementally.

6.2. Word Embeddings

Algorithms for embedding text into a vector space yield representations that can be quite beneficial in many applications, *e.g.* features for sentiment analysis. Mikolov et al. (2013b) proposed a context-based embedding called skip-gram or `word2vec`. The context of a word can be defined as a set of words before, around, or after the respective word. Their model strives to find an embedding of each word so that the representation predicts the embedding of each context word around it. Levy & Goldberg (2014) subsequently showed that the word embedding model proposed by Mikolov et al. (2013b) can be reinterpreted as matrix factorization of the *PMI* matrix constructed as follows. A word c is in context of w if it lies within the respective window of w. The PMI matrix is then calculated as

$$\text{PMI}_{w,c} = \log\left(\frac{p(w,c)}{p(w)p(c)}\right).$$

In practice the probabilities $p(w,c), p(w), p(c)$ are replaced by their empirical counterparts. Further, note that $p(w,c)$ is 0 if words c and w do not coexist in the same context, which yields $-\infty$ for PMI. Levy & Goldberg (2014) suggest using an alternative: $\text{PPMI}_{w,c} = \max\{\text{PMI}_{w,c}, 0\}$. They also suggest variations of PMI hyper parameterized by k which corresponds to the number of negative samples in the training of the original skip gram model.

We employ the binomial PCA model on the normalized count matrix (instead of the PMI), in a manner similar to the clustering approach in Section 6.1. The normalized count matrix is calculated simply as $\frac{p(w,c)}{p(w)}$, without taking logarithms. This gives us a probability matrix which has each entry between 0 and 1, and which can be factorized under the binomial model greedily as per Algorithm 2.

We empirically study the embeddings obtained by binomial factorization on two tasks – word similarity and analogies. For word similarity, we use the W353 dataset (Finkelstein et al., 2001) and the MEN data (Bruni et al., 2012). Both these datasets contain words with human assigned similarity scores. We evaluate the embeddings by their cosine similarity, and measuring the correlation with the available human ratings. The fraction of correctly answered queries are returned as the metric. For the analogy task, we use the Microsoft Research (MSR) syntactic analogies (Mikolov et al., 2013c) and the Google mixed analogies dataset (Mikolov et al., 2013a). For completing analogy $a{:}b{::}c{:}x$, the prediction is calculated as $\arg\max_x \frac{\cos(c,x)\cos(b,x)}{\cos(a,x)}$. To compute accuracy, we use the multiplication similarity metric as used by Levy & Goldberg (2014). To train the word embeddings, we use the 2013 news crawl dataset[1]. We filter out stop words, non-ASCII characters, and words occurring less than 2000 times (which yields a vocabulary of 6713). Note that since we keep only the most common words, several queries from the datasets are invalid because we do not have embeddings for words appearing in them. However, we do include them by assigning invalid queries a value of 0 and reporting the overall average over the entire dataset.

[1] http://www.statmt.org/wmt14/training-monolingual-news-crawl

Table 1. Empirical study of binomial based greedy factorization shows competitive performance of word embeddings of common words across tasks and datasets.

	W353	MEN	MSR	GOOGLE
# QUERIES	353	3000	8000	19544
SVD	0.226	0.233	0.086	0.092
PPMI	0.175	0.178	0.210	0.130
SGNS	0.223	0.020	0.052	0.002
GREEDY	0.202	0.198	0.176	0.102

Table 1 shows the empirical evaluation. SVD and PPMI are the models proposed by Levy & Goldberg (2014), while SGNS is the skipgram with negative sampling model of Mikolov et al. (2013b). We run each of these for $k = \{5, 10, 15, 20\}$ and report the best results. This shows that alternative factorizations such as our application of binomial PCA can be more consistent and competitive with other embedding methods.

Conclusion: We have connected the problem of greedy low rank matrix estimation to that of submodular optimization. Through that connection we have provided improved exponential rates of convergence for the algorithm. An interesting area of future study will be to connect these ideas to general atoms or dictionary elements.

Acknowledgements

We thank the anonymous reviewers for their helpful feedback. Research supported by William Hartwig Fellowship, NSF Grants CCF 1344179, 1344364, 1407278, 1422549, 1618689, IIS 1421729, and ARO YIP W911NF-14-1-0258.

References

Barron, Andrew R., Cohen, Albert, Dahmen, Wolfgang, and DeVore, Ronald A. Approximation and learning by greedy algorithms. *The Annals of Statistics*, 36(1):6494, Feb 2008.

Bruni, Elia, Boleda, Gemma, Baroni, Marco, and Tran, Nam-Khanh. Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pp. 136–145, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

Buhlmann, Peter and Yu, Bin. Boosting. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):6974, Dec 2009. ISSN 1939-5108.

Chen, Yudong and Wainwright, Martin J. Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees. *arXiv*, abs/1509.03025, 2015. URL http://arxiv.org/abs/1509.03025.

Collins, Michael, Dasgupta, Sanjoy, and Schapire, Robert E. A generalization of principal component analysis to the exponential family. In *Advances in Neural Information Processing Systems*. MIT Press, 2001.

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

Dudik, Miro, Harchaoui, Zaid, and Malick, Jerome. Lifted coordinate descent for learning with trace-norm regularization. In *AISTATS*, 2012.

Elenberg, Ethan R., Khanna, Rajiv, Dimakis, Alexandros G., and Negahban, Sahand. Restricted Strong Convexity Implies Weak Submodularity. *Proc. NIPS Workshop on Learning in High Dimensions with Structure*, December 2016.

Feige, Uriel. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

Finkelstein, Lev, Gabrilovich, Evgeniy, Matias, Yossi, Rivlin, Ehud, Solan, Zach, Wolfman, Gadi, and Ruppin, Eytan. Placing search in context: the concept revisited. pp. 406–414, 2001.

Gribonval, Rémi and Vandergheynst, P. On the exponential convergence of matching pursuits in quasi-incoherent dictionaries. *IEEE Trans. Inform. Theory*, 52(1):255–261, 2006.

Jaggi, Martin. Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization. In *ICML*, pp. 427–435, 2013.

Jaggi, Martin and Sulovský, Marek. A simple algorithm for nuclear norm regularized problems. In Frnkranz, Johannes and Joachims, Thorsten (eds.), *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 471–478. Omnipress, 2010.

Jain, Prateek, Netrapalli, Praneeth, and Sanghavi, Sujay. Low-rank matrix completion using alternating minimization. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pp. 665–674, 2013.

Jain, Prateek, Tewari, Ambuj, and Kar, Purushottam. On iterative hard thresholding methods for high-dimensional m-estimation. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 685–693, 2014.

Khanna, Rajiv, Tschannen, Michael, and Jaggi, Martin. Pursuits in Structured Non-Convex Matrix Factorizations. *arXiv*, 2016. 1602.04208v1.

Khanna, Rajiv, Elenberg, Ethan R., Dimakis, Alexandros G., Neghaban, Sahand, and Ghosh, Joydeep. Scalable Greedy Support Selection via Weak Submodularity. *AISTATS*, 2017.

Lacoste-Julien, Simon and Jaggi, Martin. On the Global Linear Convergence of Frank-Wolfe Optimization Variants. In *NIPS 2015*, pp. 496–504, 2015.

Lee, Kiryung and Bresler, Yoram. Corrections to "admira: Atomic decomposition for minimum rank approximation". *IEEE Trans. Information Theory*, 59(7):4730–4732, 2013.

Levy, Omer and Goldberg, Yoav. Neural word embedding as implicit matrix factorization. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27*, pp. 2177–2185. Curran Associates, Inc., 2014.

Locatello, Francesco, Khanna, Rajiv, Tschannen, Michael, and Jaggi, Martin. A unified optimization view on generalized matching pursuit and frank-wolfe. In *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

Loh, Po-Ling and Wainwright, Martin J. Regularized m-estimators with nonconvexity: Statistical and algorithmic theory for local optima. *J. Mach. Learn. Res.*, 16(1): 559–616, January 2015. ISSN 1532-4435.

Mikolov, Tomas, Chen, Kai, Corrado, Greg, and Dean, Jeffrey. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013a.

Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg S, and Dean, Jeff. Distributed representations of words and phrases and their compositionality. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 26*, pp. 3111–3119. Curran Associates, Inc., 2013b.

Mikolov, Tomas, Yih, Scott Wen-tau, and Zweig, Geoffrey. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2013)*. Association for Computational Linguistics, May 2013c.

Negahban, Sahand and Wainwright, Martin J. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *The Annals of Statistics*, 39(2):1069–1097, 2011. ISSN 00905364.

Negahban, Sahand and Wainwright, Martin J. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *JMLR*, 13, 2012.

Negahban, Sahand, Ravikumar, Pradeep, Yu, Bin, and Wainwright, Martin J. A Unified Framework for High-Dimensional Analysis of M-Estimators with Decomposable Regularizers. *Statistica Sinica*, 27(4):538–557, 2012. ISSN 0883-4237. doi: 10.1214/12-STS400.

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

Ng, Andrew Y., Jordan, Michael I., and Weiss, Yair. On spectral clustering: Analysis and an algorithm. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pp. 849–856. MIT Press, 2001.

Park, Dohyung, Kyrillidis, Anastasios, Bhojanapalli, Srinadh, Caramanis, Constantine, and Sanghavi, Sujay. Provable non-convex projected gradient descent for a class of constrained matrix optimization problems. *arXiv*, abs/1606.01316, 2016. URL http://arxiv.org/abs/1606.01316.

Rao, Nikhil, Shah, Parikshit, and Wright, Stephen. Forward backward greedy algorithms for atomic norm regularization. *IEEE Transactions on Signal Processing*, 63(21):57985811, Nov 2015. ISSN 1941-0476. doi: 10.1109/tsp.2015.2461515. URL http://dx.doi.org/10.1109/tsp.2015.2461515.

Recht, Benjamin, Fazel, Maryam, and Parrilo, Pablo A. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471501, Jan 2010. ISSN 1095-7200.

Rohde, Angelika and Tsybakov, Alexandre B. Estimation of high-dimensional low-rank matrices. *The Annals of Statistics*, 39(2):887930, Apr 2011. ISSN 0090-5364. doi: 10.1214/10-aos860. URL http://dx.doi.org/10.1214/10-aos860.

Shalev-Shwartz, S, Gonen, A, and Shamir, O. Large-scale convex minimization with a low-rank constraint. In *ICML*, 2011.

Shi, Jianbo and Malik, Jitendra. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, August 2000. ISSN 0162-8828.

Wang, Zheng, Lai, Ming-Jun, Lu, Zhaosong, Fan, Wei, Davulcu, Hasan, and Ye, Jieping. Orthogonal rank-one matrix pursuit for low rank matrix completion. *SIAM Journal on Scientific Computing*, 37(1):A488A514, Jan 2015. ISSN 1095-7197. doi: 10.1137/130934271. URL http://dx.doi.org/10.1137/130934271.