Simultaneous Feature and Feature Group Selection through Hard Thresholding

Shuo Xiang
Arizona State University
Tempe, AZ 85287, USA
shuo.xiang@asu.edu

Tao Yang
Arizona State University
Tempe, AZ 85287, USA
t.yang@asu.edu

Jieping Ye
Arizona State University
Tempe, AZ 85287, USA
jieping.ye@asu.edu

ABSTRACT

Selecting an informative subset of features has important applications in many data mining tasks especially for high-dimensional data. Recently, simultaneous selection of features and feature groups (a.k.a bi-level selection) becomes increasingly popular since it not only reduces the number of features but also unveils the underlying grouping effect in the data, which is a valuable functionality in many applications such as bioinformatics and web data mining. One major challenge of bi-level selection (or even feature selection only) is that computing a globally optimal solution requires a prohibitive computational cost. To overcome such a challenge, current research mainly falls into two categories. The first one focuses on finding suitable continuous computational surrogates for the discrete functions and this leads to various convex and nonconvex optimization models. Although efficient, convex models usually deliver sub-optimal performance while nonconvex models on the other hand require significantly more computational effort. Another direction is to use greedy algorithms to solve the discrete optimization directly. However, existing algorithms are proposed to handle single-level selection only and it remains challenging to extend these methods to handle bi-level selection. In this paper, we fulfill this gap by introducing an efficient sparse group hard thresholding algorithm. Our main contributions are: (1) we propose a novel bi-level selection model and show that the key combinatorial problem admits a globally optimal solution using dynamic programming; (2) we provide an error bound between our solution and the globally optimal under the RIP (Restricted Isometry Property) theoretical framework. Our experiments on synthetic and real data demonstrate that the proposed algorithm produces encouraging performance while keeping comparable computational efficiency to convex relaxation models.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications-Data Mining

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
KDD'14, August 24–27, 2014, New York, NY, USA.
Copyright 2014 ACM 978-1-4503-2956-9/14/08 ...$15.00.
http://dx.doi.org/10.1145/2623330.2623662.

General Terms

Algorithms

Keywords

Feature selection, supervised learning, bi-level learning, combinatorics, optimization, dynamic programming

1. INTRODUCTION

Feature selection plays a critical role in many data mining applications that handle high-dimensional data and has been one of the most active research topics in machine learning. Over the past decades, with the development of compressive sensing techniques [24, 9, 11], joint modeling of prediction and feature selection gains its popularity and draws extensive studies [34, 19, 1, 33, 31, 30]. In the meantime, it is also believed that when the data possesses certain grouping structures, selecting feature groups together with individual features can be beneficial [32, 26, 7, 17, 28]. In the literature, simultaneous selection of features and feature groups is also referred to as bi-level selection [17, 29] and we will use these two terms interchangeably throughout the paper.

Formally, we consider the following linear model in this paper: the observations $\boldsymbol{y}$ can be generated via $\boldsymbol{y} = \boldsymbol{A}\bar{\boldsymbol{x}} + \boldsymbol{\epsilon}$, where each row of $\boldsymbol{A} \in \mathbb{R}^{n\times p}$ contains a sample and $\boldsymbol{\epsilon}$ represents the noise. $\bar{\boldsymbol{x}} \in \mathbb{R}^p$ is the parameter in this linear regression setting. In addition, the elements of $\bar{\boldsymbol{x}}$ are divided into $|G|$ mutually exclusive feature groups with G_i denoting the indices that belong to the ith group. The target is to find an accurate estimator $\boldsymbol{x}$ for $\bar{\boldsymbol{x}}$ based on our observations of $\boldsymbol{A}$ and $\boldsymbol{y}$. In the meantime, we expect the solution to yield sparsity in both feature level and group level, i.e., only a small number of features and feature groups are selected and thus a large potion of the elements/groups of $\boldsymbol{x}$ admits the value of zero. Mathematically, we formulate our problem in Eq. (1) below, where we attempt to find the best solution (in the sense of least squares) among all candidates containing no more than s_1 nonzero values and taking up to at most s_2 feature groups:

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 \\ \text{subject to} \quad & \sum_{j=1}^{p} I(|x_j| \neq 0) \leq s_1 \\ & \sum_{j=1}^{|G|} I(\|\boldsymbol{x}_{G_j}\|_2 \neq 0) \leq s_2. \end{aligned} \tag{1}$$

Unfortunately, computing an optimal solution of the above problem requires enumerating all elements in the feasible set and thus incurs a prohibitive cost. A natural and popular approach is to replace the discrete constraints in Eq. (1) by their continuous computational surrogates. Sparse group lasso [14] applies the classical ℓ_1-relaxation on both feature-level and group-level and the resulting convex optimization problem can be efficiently solved. To enhance the quality of approximation, nonconvex relaxations using DC programming [28] and nonconvex penalties such as group MCP [7] and group bridge [16] are introduced, with extra computation effort. The connections between various convex and nonconvex bi-level learning models are investigated in the literature [29]. On the other hand, instead of finding suitable continuous surrogates, computing a local solution of the discrete optimization problem directly also receives plenty of attention. The iterative hard thresholding (IHT) [5, 6], orthogonal matching pursuit [25] and group orthogonal matching pursuit [20] all fall into this category. Although the optimization is by nature nonconvex, the efficiency of these algorithms is usually comparable (if not better) to that of convex relaxation models. However, to the best of our knowledge, these algorithms are proposed for feature selection only or group selection only. Whether they can be extended to handle bi-level selection properly and efficiently has not been much explored.

In this paper, we fulfill such a gap by introducing a hard thresholding model that is capable of bi-level selection. Our main contributions are: (1) we propose a novel bi-level selection model and show that the key combinatorial problem admits a globally optimal solution using dynamic programming; (2) we provide an error bound between our solution and the globally optimal under the RIP (Restricted Isometry Property) theoretical framework [9, 8]. We have evaluated the proposed algorithm on synthetic and real data. Results show that the proposed algorithm demonstrates encouraging performance while keeping comparable computational efficiency as convex relaxation models.

The remaining of the paper is organized as follows: We present our algorithm for Problem (1) and discuss different variants in Section 2. In Section 3, we investigate a key subproblem in our method and propose a dynamic programming algorithm that finds an optimal solution. The convergence property of the overall optimization framework is discussed in Section 4 and we present extensive empirical evaluation in Section 5. Section 6 concludes the paper and lists our plan of future work. For notations, we mainly follow the symbols introduced in Eq. (1), i.e., $\boldsymbol{A}$ stands for the design (sample) matrix, $\boldsymbol{y}$ is the response, $\boldsymbol{x}_{G_i}$ represents the regression model restricted on the ith group and f denotes the objective function.

2. OPTIMIZATION ALGORITHMS

Motivated by the iterative hard thresholding algorithm for ℓ_0-regularized problems [6] and the recent advances on nonconvex iterative shrinkage algorithm [15], we adopt the Iterative Shrinkage and Thresholding Algorithm (ISTA) framework and propose the following algorithm for solving Problem (1):

In the proposed algorithm above, f denotes the objective function and the "SGHT" in Algorithm 1 stands for the following **Sparse Group Hard Thresholding (SGHT)** problem with $\boldsymbol{v}$ as the input:

Algorithm 1 ISTA with Sparse Group Hard Thresholding

Input: $\boldsymbol{A}$, $\boldsymbol{y}$, s_1, s_2, $\eta > 1$
Output: solution $\boldsymbol{x}$ to Problem (1)
1: Initialize $\boldsymbol{x}^0$.
2: **for** $m \leftarrow 1, 2, \cdots$ **do**
3: Initialize L
4: **repeat**
5: $\boldsymbol{x}^m \leftarrow \text{SGHT}(\boldsymbol{x}^{m-1} - \frac{1}{L}\nabla f(\boldsymbol{x}^{m-1}))$
6: $L \leftarrow \eta L$
7: **until** line search criterion is satisfied
8: **if** the objective stops decreasing **then**
9: **return** $\boldsymbol{x}^m$
10: **end if**
11: **end for**

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{v}\|_2^2 \\ \text{subject to} \quad & \sum_{j=1}^{p} I(|x_j| \neq 0) \leq s_1 \\ & \sum_{j=1}^{|G|} I(\|\boldsymbol{x}_{G_j}\|_2 \neq 0) \leq s_2. \end{aligned} \tag{2}$$

Like most ISTA-based optimization algorithms, it is of critical importance that we can compute the projection step accurately and efficiently. In our case, the key part is exactly the SGHT problem. Although there are well established results on hard thresholding algorithms for ℓ_0-regularization, adding one more constraint on group cardinality greatly complicates the problem and requires deeper analysis. We will present detailed discussion on how to compute an optimal solution to this problem efficiently in the next section. Before that, we first introduce several possible variants of the proposed method. Notice that the target of Algorithm 1 is a nonconvex optimization problem. Different strategies for initialization and step-size may not only provide different convergence behavior, but also lead to a completely different solution. We consider three aspects in this paper: step-size initialization, line search criterion and acceleration option.

2.1 Step-size Initialization

To provide an initial value of the step-size (Line 6. in Algorithm 1), we consider two strategies: a constant value and the Barzilai-Borwein (BB) method [2]. The BB method essentially finds the best multiple of identity matrix to approximate the Hessian matrix such that the least squares error of the secant equation is minimized, i.e., L^k is initialized to

$$\begin{aligned} \alpha^k &= \arg\min_{\alpha} \|\alpha(x^k - x^{k-1}) - (\nabla f(x^k) - \nabla f(x^{k-1}))\| \\ &= \frac{(\Delta g)^T(\Delta x)}{\|\Delta x\|^2} \end{aligned}$$

with a safeguard bound, where $\Delta g = \nabla f(x^k) - \nabla f(x^{k-1})$ and $\Delta x = x^k - x^{k-1}$. In this paper, we set $L^k = \max(1, \alpha^k)$.

2.2 Line Search Criterion

We consider two line search termination criteria in this paper, which we name as Lipschiz criterion and sufficient decrease criterion. Specifically the Lipschiz criterion finds

the smallest L that the following inequality is satisfied:

$$f(x^k) \leq f(x^{k-1}) + \langle \nabla f(x^{k-1}), x^k - x^{k-1} \rangle + \frac{L}{2}\|x^k - x^{k-1}\|_2^2. \quad (3)$$

On the other hand, the sufficient decrease criterion aims to find the smallest L such that:

$$f(x^k) \leq f(x^{k-1}) - \frac{L\delta}{2}\|x^k - x^{k-1}\|_2^2. \quad (4)$$

Inequality (3) is the standard way for ℓ_1-regularized optimization [3] and is applied extensively in structured sparse learning [19]. Inequality (4) and its variants are favored by most of the recent investigations on nonconvex regularized problems [4, 27, 15].

2.3 Acceleration Option

The ISTA framework has been shown to possess a convergence rate of $O(1/k)$ for a class of ℓ_1-regularized/constrained optimization problems and can be further improved to $O(1/k^2)$ via adding a carefully designed search point [21, 3]. However, whether the same strategy still works or makes the optimization diverge in the regime of nonconvex optimization remains unknown. In this paper we consider both of them and retain the notation of FISTA [3] to denote the ISTA with the acceleration trick. See Algorithm 2 for more detail about our FISTA.

Algorithm 2 FISTA with Sparse Group Hard Thresholding

Input: $\boldsymbol{A}$, $\boldsymbol{y}$, s_1, s_2, $\eta > 1$
Output: solution $\boldsymbol{x}$ to Problem (1)

```
Initialize x^{-1}, x^0, α^{-1} ← 0, α^0 ← 1
for m ← 1, 2, ··· do
    β^m ← (α^{m-2} − 1) / α^{m-1}
    u^m ← x^{m-1} + β^m (x^{m-1} − x^{m-2})
    Initialize L
    repeat
        x^m ← SGHT(u^m − (1/L) ∇f(u^m))
        L ← ηL
    until line search criterion is satisfied
    if the objective stops decreasing then
        return x^m
    end if
end for
```

Table 1: Specific settings for each variant considered in the paper. The last two columns denote the Lipschiz and sufficient decrease line search criterion respectively.

VARIANTS	FISTA	ISTA	BB	CONST	LIPS	DEC
ISTA		✓	✓			✓
ISTA-L		✓	✓		✓	
FISTA	✓		✓		✓	
FISTA-C	✓			✓	✓	

Table 1 summaries different variants we consider in this paper. All these variants will be examined in our experiments. We conclude this section by presenting several additional features of the proposed algorithm.

Remark 1. One significant advantage of adhering to the discrete model is that incorporating prior knowledge about the grouping structure is quite straight-forward. Remember that the two parameters in our model are just the upper-bound of features and feature groups respectively. In addition, model selection procedures such as cross-validation can be greatly facilitated since we only need to consider integer values, which are often quite small in real-world applications. On the contrary, the regularizers in most of the existing works are real-valued and may not provide much insights for parameter-tuning.

Remark 2. Although we consider our bi-level learning model in a linear regression setting, the technique can be readily extended to more general problems by choosing appropriate loss functions. Particularly, in order to extend our model to classification tasks, the widely-used logistic loss function can be applied instead of the least squares function in Eq. (1) and the proposed Algorithm 1 can be applied by changing the procedure that computes the gradient. In general, the proposed model can be extended to any convex loss functions with a simple gradient computation.

3. OPTIMAL SOLUTION OF SGHT

In this section, we show how to solve the SGHT problem in Eq. (2) efficiently using dynamic programming. Before presenting our algorithm, we first explore some key properties of Problem (2). As highlighted previously, the major challenge comes from the two coupled constraints. Therefore, we first consider the special case where only one of the two constraints is present. Some straight-forward analysis leads to the following results:

LEMMA 1. *If only the cardinality constraint is present, the optimal solution of Problem* (2) *can be obtained by setting the $p - s_1$ smallest (in absolute value) elements of $\boldsymbol{v}$ to zero. Similarly for group cardinality constraint, it suffices to find the $|G| - s_2$ smallest groups (in ℓ_2-norm) and set them to zero.*

Based on Lemma 1, it is also easy to verify that for any optimal solution $\boldsymbol{x}^*$ of Problem (2), each element x_i^* is either equal to v_i or zero, where the subscript i denotes the ith element of the vector. Therefore we have the following proposition providing an equivalent but discrete characterization of the original SGHT problem:

PROPOSITION 1. *Finding the optimal solution of problem* (2) *is equivalent to the following* ***Sparse Group Subset Selection (SGSS)*** *problem:*

Given a set S on which a nonnegative value function f is defined. $C = \{C_1, C_2, \cdots, C_{|G|}\}$ is a collection of disjoint subsets of S such that $S = \bigcup_{i=1}^{|G|} C_i$. Find a subset $S' \subset S$ with the maximum value such that the cardinality of S is no more than s_1 and S' has nonempty intersections with at most s_2 elements from C. The value of a subset is defined as the summation of all the values of its elements.

We claim that the SGHT has an optimal solution if and only if we can find an optimal solution for the SGSS problem. We provide a one-way reduction (the "if" part) here. The other way is almost identical. The original SGHT problem can be reduced to SGSS by simply setting $S = \{1, 2, \cdots, p\}$ with the value function defined as $f(i) = v_i^2$ for all $1 \leq i \leq p$ and $C_i = G_i$ for all $1 \leq i \leq |G|$. Suppose S' is the optimal solution of SGSS. Then the optimal solution of SGHT can

be readily obtained via:

$$x^* = \begin{cases} v_i & \text{if } i \in S' \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

In the sequel, we will focus on the SGSS problem and provide an efficient algorithm to compute its globally optimal solution. The term cardinality and group cardinality are used to characterize the size of S' and the number of elements from C with which S' has a nonempty intersection, respectively.

Let $T(i, j, k)$ denote the maximum value we can obtain by choosing a subset S', whose cardinality is no more than k and group cardinality is at most j. In addition, S' is only allowed to have nonempty intersection with $C_1, C_2, \cdots, C_i$. Therefore T is in essence a three-dimensional table of size $(|G|+1) \times (s_2+1) \times (s_1+1)$ (the table is zero-indexed). It is easy to verify that, if we are able to compute all the values in table T correctly, the maximum value one can get in the SGSS problem is given by $T(|G|, s_2, s_1)$.

Next we propose a dynamic programming algorithm to compute the table T. The motivation behind our method is the existence of optimal substructure and overlapping subproblems [18], two major ingredients for an efficient dynamic programming algorithm. More specifically, when we try to compute $T(i, j, k)$, the optimal solution must fall into one of the two situations: whether the C_i is selected or not. If not, we can simply conclude that $T(i, j, k) = T(i-1, j, k)$. On the other hand, if C_i is selected, we need to determine how many elements from C_i are included in the optimal solution. Suppose the optimal solution takes t elements from C_i, then we must have $T(i, j, k) = T(i-1, j-1, k-t) + CH(i, t)$, where $CH(i, t)$ denotes the maximum value one can get from choosing t elements out of C_i. The optimal t can be computed via enumeration. To sum up, the computation of $T(i, j, k)$ can be written in the following recursive form:

$$T(i, j, k) = \max \begin{cases} T(i-1, j, k) \\ \max\limits_{1 \le t \le \min(k, |G_i|)} T(i-1, j-1, k-t) + CH(i, t). \end{cases}$$

It is clear from above that $T(i, j, k)$ can be computed using only the values in the table T with smaller indices. Therefore we can compute each element of the table T in increasing order for each index; see Figure 1 for more detail. In addition, to further reduce the complexity, function $CH(i, t)$ can be precomputed before the dynamic programming process. We present the detailed description of the proposed method in Algorithm 3. From table T, we are able to calculate the minimum objective value of the SGHT problem, which is exactly $\frac{1}{2}(\|\boldsymbol{v}\|_2^2 - T(|G|, s_2, s_1))$. In order to calculate the optimal solution $\boldsymbol{x}^*$, all we need to know is the indices of selected elements in S and the optimal solution can be constructed through Eq. (5). We compute such information by adding one table P (stands for path) in the proposed algorithm. Specifically, $P(i, j, k) = 0$ means the C_i is not selected in the computation of $T(i, j, k)$. Otherwise we set

$$P(i, j, k) = \arg \max_{1 \le t \le \min(k, |G_i|)} T(i-1, j-1, k-t) + CH(i, t),$$

which is just the number of selected features in the ith group (C_i) in the optimal solution. To recover the indices of all the selected elements, we will start from $P(|G|, s_2, s_1)$ with a backtracking procedure and record the number of selected elements in each group. Algorithm 4 provides a formal description of this process. It accepts the table P as input and returns the *cnt* table which contains the number of selected elements in each group. Finally computing the optimal $\boldsymbol{x}^*$ only amounts to keeping the top selected elements for each group and setting the remains to zero.

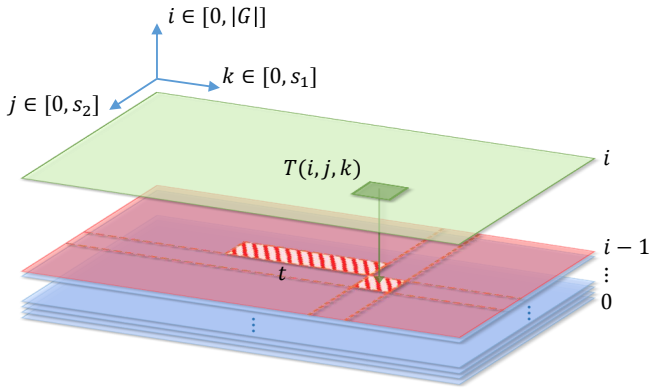

Figure 1: Illustration of the order of computation for each element in T. While computing $T(i, j, k)$, we only need values in those red squares, which are located in the previous rectangle (in terms of i-axis) and of equal or smaller coordinates on axes j and k. Therefore the computation can be naturally carried out in three nested loops, one for each axis respectively.

We analyze the time complexity of our proposed algorithm as follows. Notice that the time needed to precompute the table CH is give by:

$$O(\sum_{i=1}^{|G|} |G_i| \log(|G_i|)) = O(p \log p),$$

the dynamic programming part for computing both T and P takes

$$O(\sum_{i=1}^{|G|} s_2 s_1 |G_i|) = O(s_1 s_2 \sum_{i=1}^{|G|} |G_i|) = O(p s_1 s_2),$$

and the backtracking needs clearly $O(|G|)$ operations. Therefore the overall time complexity is

$$O(p(s_1 s_2 + \log p) + |G|) = O(s_1 s_2 p + p \log p).$$

When the number of features and feature groups selected is small, the SGHT problem can be solved efficiently.

4. CONVERGENCE ANALYSIS

In this section, knowing that the key SGHT sub-problem can be efficiently computed, we assess the quality of the solution produced by the overall optimization procedure (Algorithm 1). Specifically, since the constraints of Eq. (1) are nonconvex and only a local minimum can be found through our proposed method, we are interested in studying how close (in terms of Euclidean distance) the obtained solution to the optimal solution of the optimization problem (1). Although we are not aware of the optimal solution, the bound

Algorithm 3 Dynamic programming algorithm for SGSS

Input: S, $C = \bigcup_{i=1}^{|G|} C_i$, s_1, s_2
Output: T, P

```
T ← 0, CH ← 0, P ← 0
for i = 1 to |G| do
    sort C_i in decreasing order of magnitude
    for t = 1 to |G_i| do
        CH(i,t) ← CH(i,t−1) + C_i(t)
    end for
end for
for i = 1 to |G| do
    for j = 1 to s_2 do
        for k = 1 to s_1 do
            T(i,j,k) ← T(i−1,j,k)
            for t = 1 to G_i do
                w ← T(i−1,j−1,k−t) + CH(i,t)
                if w > T(i,j,k) then
                    T(i,j,k) = w
                    P(i,j,k) = t
                end if
            end for
        end for
    end for
end for
```

Algorithm 4 Linear backtracking algorithm for finding the number of selected elements in each group

Input: P , s_1, s_2
Output: cnt

```
j ← s_2, k ← s_1
for i = |G| downto 1 do
    cnt(i) ← P(i,j,k)
    if cnt(i) > 0 then
        j ← j − 1
        k ← k − cnt(i)
    end if
end for
```

between our solution and the optimal one can be analyzed under the theoretical framework of restricted isometry property (RIP) [9]. A matrix $\boldsymbol{A} \in \mathbb{R}^{n\times p}$ is said to satisfy the RIP property with constant δ_s if the following property holds for any s-sparse vector $\boldsymbol{x}$, i.e., $\|\boldsymbol{x}\|_0 \leq s$:

$$(1-\delta_s)\|\boldsymbol{x}\|_2^2 \leq \|\boldsymbol{A}\boldsymbol{x}\|_2^2 \leq (1+\delta_s)\|\boldsymbol{x}\|_2^2.$$

The RIP constant essentially assesses the extent to which the given matrix resembles an orthogonal matrix and theoretical analyses often require certain upperbound on the RIP constant. It is easy to see that δ_s is non-decreasing w.r.t s and a smaller value of δ_s indicates more rigid conditions we require from $\boldsymbol{A}$. In order to apply the RIP based analysis for our method, a group-RIP constant is introduced to incorporate the group structure. Matrix $\boldsymbol{A}$ has a group-RIP constant δ^g if for any vector $\boldsymbol{x}$ that spans no more than g groups, i.e., $\sum_{j=1}^{|G|} I(\|\boldsymbol{x}_{G_j}\|_2 \neq 0) \leq g$, the following relation are satisfied:

$$(1-\delta^g)\|\boldsymbol{x}\|_2^2 \leq \|\boldsymbol{A}\boldsymbol{x}\|_2^2 \leq (1+\delta^g)\|\boldsymbol{x}\|_2^2.$$

Our next result provides an error bound between an optimal solution of Problem (1) and the solution produced by our proposed Algorithm 1 with L fixed to 1.

THEOREM 1. *Let $\boldsymbol{x}^*$ be a globally optimal solution of Problem (1) and $\boldsymbol{x}^k$ be the solution we obtain after the kth iteration in Algorithm 1 with $L = 1$. If $c_1 < \frac{1}{2}$, the following result holds:*

$$\|\boldsymbol{x}^k - \boldsymbol{x}^*\|_2 \leq (2c_1)^k\|\boldsymbol{x}^0 - \boldsymbol{x}^*\|_2 + \frac{2\sqrt{1+c_2}}{1-2c_1}\|\boldsymbol{e}^*\|_2,$$

where $\boldsymbol{e}^* = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^*$, $c_1 = \min\{\delta_{3s_1}, \delta^{3s_2}\}$, $c_2 = \min\{\delta_{2s_1}, \delta^{2s_2}\}$. In addition, if $c_2 < \frac{1}{4}$, it is also true that:

$$\|\boldsymbol{x}^k - \boldsymbol{x}^*\|_2 \leq (4c_2)^k\|\boldsymbol{x}^0 - \boldsymbol{x}^*\|_2 + \frac{2\sqrt{1+c_2}}{1-4c_2}\|\boldsymbol{e}^*\|_2.$$

Theorem 1 clearly shows that the parameter estimation error of the proposed algorithm decreases linearly (with coefficient of $2c_1$ or $4c_2$) till a fixed error term is met. In addition, such an error term is proportional to the prediction error of the optimal solution of Problem (1). The proof of Theorem 1 mainly utilizes the technique in [12] and the details are left in the Appendix. We provide an illustrative example of the convergence procedure in Figure 2: if the assumptions on the (group) RIP constant hold, the sequence generated by running our algorithm is guaranteed to converge into a region centered at $\boldsymbol{x}^*$ with radius at most $c\|\boldsymbol{e}^*\|_2$, where c is a constant. As we can observe from Figure 2 and Theorem 1, the difference between the unknown globally optimal solution of Problem (1) and ours is upperbounded by a multiple of the underlying error term $\|\boldsymbol{e}^*\|_2$. In addition, such a difference cannot be canceled unless we have $\boldsymbol{e}^* = 0$, in which case Theorem 1 essentially states that our method admits a linear convergence rate [22].

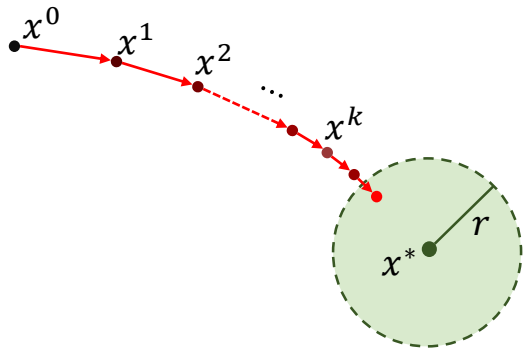

Figure 2: Illustration of the convergence behavior of the proposed algorithm. The parameter estimation error decreases linearly before entering into a region centered at $\boldsymbol{x}^*$ with radius proportional to the prediction error of $\boldsymbol{x}^*$.

5. EXPERIMENTS

5.1 Evaluation of SGHT

Recall that solving SGHT (Problem (2)) accurately and efficiently is the key to our optimization procedure (Algorithm 1). We have theoretically analyzed the correctness and time complexity of our method in Section 3. In this part, we present empirical studies on the efficiency of our proposed Algorithm 3. As we have analyzed previously, three factors including the number of candidate features, the number of

selected groups and the number of selected features determine the time complexity. We conduct the evaluation in four different scenarios, each of which demonstrates the relationship between the running time and some particular factors while keeping other factors unchanged. Specific settings are listed in Table 2.

Table 2: Experiment setup for evaluation of SGHT

Fixed variable	# Group	# Feature	s_1	s_2
Scenario 1	✓			
Scenario 2		✓		
Scenario 3		✓	✓	✓
Scenario 4	✓	✓		

- **Scenario 1. Varying number of features p with incremental candidate set.** We vary the number of features p from $1,000$ to $5,000,000$. The number of groups is fixed to 100 in this case, i.e., $|G| = 100$. s_2 is set to 20%, 40% and 60% of the total number of groups respectively and the value of s_1 is set to $5s_2$, i.e., we want to approximately select 5 features per group.
- **Scenario 2. Varying number of groups $|G|$ with incremental candidate set.** p is fixed to $1,000,000$ and G is chosen from the set of $\{10, 50, 100, 150, 200\}$. The value of s_1 and s_2 is set according to the same strategy in Scenario 1.
- **Scenario 3. Varying number of groups $|G|$ with fixed candidate set.** We conduct this evaluation in order to verify our theoretical result that the number of groups $|G|$ is not a dominating factor of time complexity. Specifically we fix the value of p to $1,000,000$ and choose $|G|$ from $\{50, 100, 500, 1000, 5000, 10000\}$. s_1 and s_2 are fixed as 50 and 5 respectively.
- **Scenario 4. Incremental candidate set with fixed number of groups and features.** In this case, $1,000,000$ variables are partitioned into 100 groups of equal size. We attempt to select $10\% \sim 60\%$ of all the groups and approximately 20 features per group.

Figure 3 demonstrates the running time (in seconds) of our SGHT algorithm of all four scenarios. Specifically, the nearly flat curve in our third experiment corroborates with the theoretical result that the number of groups is not a major factor of the time complexity. In other cases, our algorithm exhibits its capability of handling large-scale applications. Particularly, when only a small number of features and feature groups are wanted, as is the common situation in high-dimensional variable selection, our algorithm is capable of computing a globally optimial solution for SGHT with a performance competitive to its convex computational surrogate such as the soft-thresholding [10].

5.2 Evaluation of Convergence

We study the convergence behavior of different implementations of our discrete optimization approach proposed in Section 1. The evaluation is carried out on a collection of randomly generated data sets $(\boldsymbol{A}, \boldsymbol{y})$. Specifically, we generate $\boldsymbol{A} \in \mathbb{R}^{n\times p}, \boldsymbol{y} \in \mathbb{R}^n$, where the values of n and p are chosen from the following set:

$$\{(100, 2000), (100, 5000), (1000, 20000), (1000, 50000)\}.$$

All of the p features are partitioned into groups of size 100. The value of s_2 is selected from $\{0.1|G|, 0.2|G|\}$, i.e., we select 10% and 20% groups. s_1 is set to $5s_2$, which leads to the effect of within-group sparsity.

For all of the variants, we terminate the programs when either the relative change of objective value in two consecutive iterations or the gradient of the objective is less than a given threshold. The objective values of up to the first 100 iterations as well as the running time for each variant are reported in Figure 4. The results demonstrate the effect of BB to initialize the step-size. Both ISTA with lipschiz line search criterion (blue in Figure 4) and FISTA (black in Figure 4) deliver superior performance, particularly for large data sets and large number of selected groups/features.

5.3 Simulation Results

We examine the proposed bi-level method on synthetic data which consist of both group selection and bi-level variable selection. The data generation follows the procedures recommended in the literature [32, 29]: the data set is generated via the linear model $\boldsymbol{y} = \boldsymbol{A}\bar{\boldsymbol{x}} + \boldsymbol{\epsilon}$, where both of the design matrix $\boldsymbol{A} \in \mathbb{R}^{100\times 200}$ and the noise term $\boldsymbol{\epsilon}$ follow a normal distribution. The ground truth $\bar{\boldsymbol{x}}$ is partitioned into 20 groups of equal size. In addition, two kinds of grouping structure are considered in this experiment; see Figure 5 for more detail. The goal is to obtain an accurate (in terms of least squares) estimator of $\bar{\boldsymbol{x}}$ that also preserves the grouping structure, given only $\boldsymbol{A}$ and $\boldsymbol{y}$.

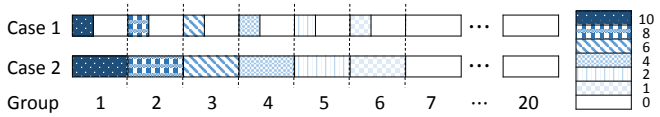

Figure 5: Illustraion of the grouping effect in the ground truth model $\bar{\boldsymbol{x}}$. Both cases include redundant groups (group 7 to group 20). In addition, the first case contains a bi-level sparsity. The values within each group are identical, as shown in the color map.

State-of-the-art bi-level feature learning algorithms, including the convex sparse group lasso, two fractional models [29] (frac$(1, 2)$ for bi-level variable selection and frac$(2, 1)$ for group selection) and DC approximation approach [28], are included for comparison. It is worth mentioning that the DC approach deals with exactly the same formulation as ours but resort to using continuous computational surrogate. In addition, we also include orthogonal matching pursuit (OMP) and group orthogonal matching pursuit (gOMP) in the experiments as they provide baseline results for discrete optimization approach. For both fractional models, we choose 5 regularizers from the interval $[10^{-8}, 10^2]$. For DC approach and our method, s_2 is selected from $\{2, 4, 6, 8, 10\}$ and s_1 is chosen from the set of $\{2s_2, 4s_2, 6s_2, 8s_2, 10s_2\}$. Since the parameters of OMP and gOMP are just the number of selected features and feature groups respectively, we set $\{6, 12, 18, \cdots, 60\}$ as the candidate parameter set for OMP and similarly $\{2, 4, 6, \cdots, 10\}$ for gOMP. Five-fold cross-validation is carried out to choose the best parameter for each method. The tuned models are then tested on an i.i.d testing set. Following the setups in previous work [7, 28], the

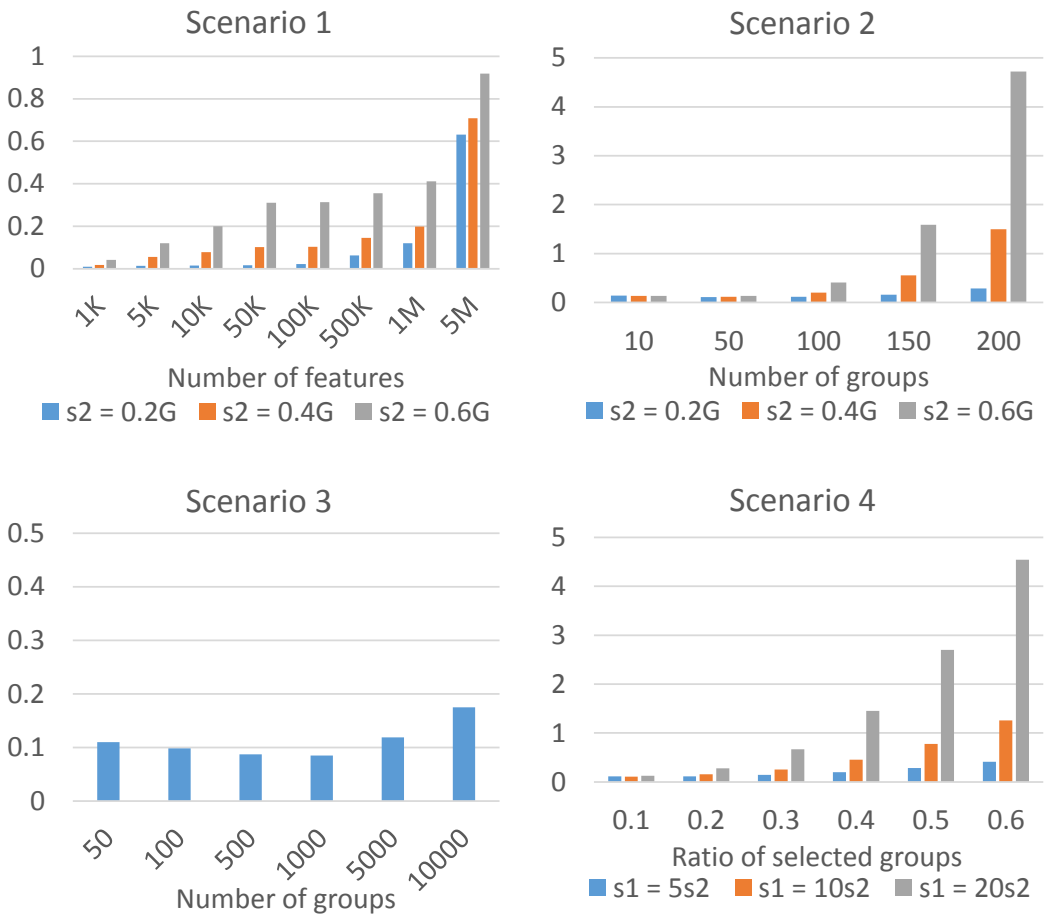

Figure 3: Running time (in seconds) of Algorithm 3 in different scenarios.

number of selected groups/features, the number of false positive selections and false negative selections and the running time (in seconds) are reported in Table 3. We can observe that the approaches with discrete parameters (OMP, gOMP, DC approach and our method) deliver more accurate estimation on the number of groups and features, compared to regularization-based approaches. Particularly, our method demonstrates the best performance in the bi-level selection tasks and is second only to gOMP in the scenario of group selection. The low false positive rate means that redundant groups are effectively screened. However, this could lead to a relatively high but still reasonable false negative rate. Such a phenomenon is also observed in existing work [7]. As of efficiency, it is expected that OMP and gOMP are the most efficient methods due to their cheap and small number of iterations. Among others, our method requires the least amount of running-time. In addition, the DC approach, which needs to refine the continous surrogate within each iteration, requires the most computational effort (nearly twice of the time of our method).

5.4 Real-world Applications

We conclude the experiment section with a study on the Boston Housing data set [13]. The original data set is used as a regression task which contains 506 samples with 13 features. Furthermore, to take into account the non-linear relationship between variables and response, up to third-degree polynomial expansion is applied on each feature, as suggested in previous works [23]. Specifically, for each variable x, we record x, x^2 and x^3 in the transformed data and gather them into one group. We randomly take 50% of the data as the training set and leave the rest for testing. The parameter settings for each method follow the same spirit in our last experiment and are properly scaled to fit this data set. We fit a linear regression model on the training data and report the number of selected features, feature groups as well as the mean squared error (MSE) on the testing set in Table 4. Five-fold cross validation is adopted for parameter tuning and all the results are averaged over 10 replications. We can observe from the table that our method shows the best prediction results with the least amount of features and feature groups.

6. CONCLUSIONS

In this paper, we study the problem of simultaneous feature and feature group selection. Unlike existing methods which are based on continuous computational surrogate for the discrete selection problem, we focus on the discrete model directly. Systematic investigations are carried out on optimization algorithms, convergence property as well as empirical evaluations. The proposed model delivers superior performance in both group selection and bi-level vari-

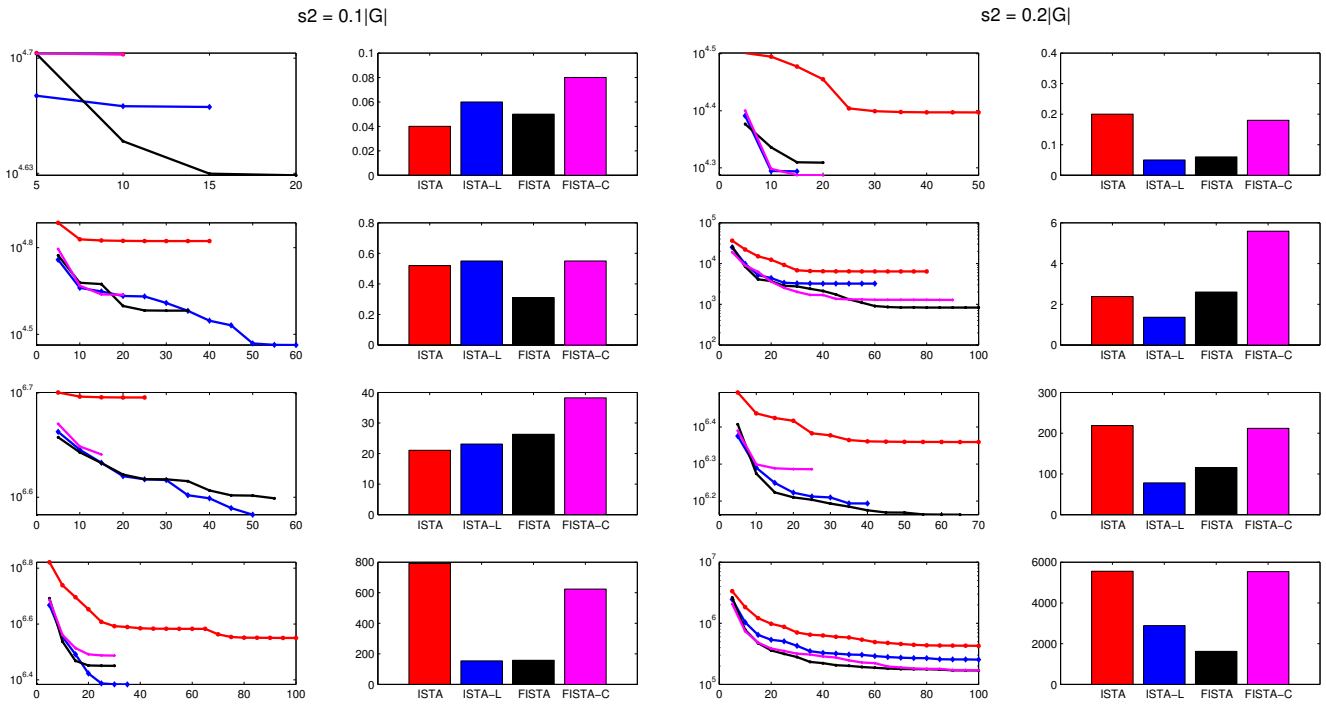

Figure 4: Convergence results of different variants of the proposed discrete optimization approach on synthetic data, where ISTA-L and FISTA-C stand for ISTA with Lipschiz line search criterion and FISTA with const step-size initialization. All the algorithms are evaluated on four data sets, from top to bottom, of which the size of $\boldsymbol{A}$ is $(100, 2000)$, $(100, 5000)$, $(1000, 20000)$ and $(1000, 50000)$ respectively. The number of selected group (s_2) is chosen from $0.1|G|$ and $0.2|G|$ and the corresponding results are listed from left to right. For each parameter setting, we report the objective values up to 100 iterations (the lines) as well as the running time in second (the histograms).

Table 4: Comparison of performance on the Boston Housing data set. All the results are averaged over 10 replications.

Methods	# Group	# Feature	mse
sgLasso	7.10	20.30	2603.50
frac$(1, 2)$	9.30	16.10	8485.12
frac$(2, 1)$	9.60	28.80	8530.00
OMP	4.30	6.00	8089.91
gOMP	4.20	12.00	8924.55
DC	2.70	5.20	8322.14
sght	2.10	3.00	545.27

able selection settings and possesses significant advantage on efficiency, particularly when only a small number of features and feature groups are demanded. In addition, due to the discrete parameters, model selection procedures such as parameter tuning can be greatly facilitated. We plan to extend our method to more challenging biomedical applications, particularly those with block-wise missing data.

7. ACKNOWLEDGMENTS

This work was supported in part by NIH (R01 LM010730) and NSF (IIS-0953662, MCB-1026710, and CCF-1025177).

8. REFERENCES

[1] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. *Convex Optimization with Sparsity-Inducing Norms*. 2010.

[2] J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.

[3] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[4] E. G. Birgin, J. M. Martínez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, 10(4):1196–1211, 2000.

[5] T. Blumensath and M. E. Davies. Iterative thresholding for sparse approximations. *Journal of Fourier Analysis and Applications*, 14(5-6):629–654, 2008.

[6] T. Blumensath and M. E. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265–274, 2009.

[7] P. Breheny and J. Huang. Penalized methods for bi-level variable selection. *Statistics and its interface*, 2(3):369–380, 2009.

Table 3: Comparison of performance on synthetic data. NO, FP and FN denote number, false positive number and false negative number respectively. All the results are averaged over 10 replications.

Methods	Bi-level Selection (case 1)							Group Selection (case 2)						
	Groups			Features			Time	Groups			Features			Time
	NO.	FP	FN	NO.	FP	FN		NO.	FP	FN	NO.	FP	FN	
sgLasso	19.10	13.10	0.00	93.30	75.30	0.00	10.4	16.70	10.80	0.10	167.00	108.00	1.00	12.2
frac$(1,2)$	8.90	2.90	0.00	59.70	41.70	0.00	15.7	8.30	3.20	0.90	59.00	19.90	20.90	29.9
frac$(2,1)$	8.60	2.80	0.20	86.00	68.60	0.60	25.9	7.50	1.70	0.20	75.00	17.00	2.00	19.8
OMP	8.40	3.00	0.60	21.00	5.90	2.90	1.6	4.80	1.80	3.00	7.20	1.90	54.70	1.6
gOMP	3.80	0.00	2.20	38.00	26.60	6.60	0.85	4.20	0.00	1.80	42.00	0.00	18.00	0.85
DC	7.70	2.00	0.30	33.20	16.20	1.00	34.3	5.60	2.00	2.40	33.90	7.00	33.10	35.6
sght	5.20	0.00	0.80	19.60	4.20	2.60	17.4	5.60	1.50	1.90	51.60	12.50	20.90	16.3

[8] E. J. Candes. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346(9):589–592, 2008.

[9] E. J. Candes and T. Tao. Decoding by linear programming. *Information Theory, IEEE Transactions on*, 51(12):4203–4215, 2005.

[10] D. Donoho. De-noising by soft-thresholding. *Information Theory, IEEE Transactions on*, 41(3):613–627, 2002.

[11] D. L. Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306, 2006.

[12] S. Foucart. Sparse recovery algorithms: sufficient conditions in terms of restricted isometry constants. In *Approximation Theory XIII: San Antonio 2010*, pages 65–77. Springer, 2012.

[13] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[14] J. Friedman, T. Hastie, and R. Tibshirani. A note on the group lasso and a sparse group lasso. *Arxiv preprint arXiv:1001.0736*, 2010.

[15] P. Gong, C. Zhang, Z. Lu, J. Huang, and J. Ye. A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. In *The 30th International Conference on Machine Learning (ICML)*, pages 37–45, 2013.

[16] J. Huang, S. Ma, H. Xie, and C. Zhang. A group bridge approach for variable selection. *Biometrika*, 96(2):339–355, 2009.

[17] J. Huang, T. Zhang, et al. The benefit of group sparsity. *The Annals of Statistics*, 38(4):1978–2004, 2010.

[18] C. E. Leiserson, R. L. Rivest, C. Stein, and T. H. Cormen. *Introduction to algorithms*. The MIT press, 2001.

[19] J. Liu, S. Ji, and J. Ye. *SLEP: Sparse Learning with Efficient Projections*. Arizona State University, 2009.

[20] A. C. Lozano, G. Swirszcz, and N. Abe. Group orthogonal matching pursuit for variable selection and prediction. In *NIPS'09-23 th Annual Conference on Neural Information Processing Systems*, 2009.

[21] Y. Nesterov et al. Gradient methods for minimizing composite objective function, 2007.

[22] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2000.

[23] G. Swirszcz, N. Abe, and A. C. Lozano. Grouped orthogonal matching pursuit for variable selection and prediction. In *Advances in Neural Information Processing Systems*, pages 1150–1158, 2009.

[24] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pages 267–288, 1996.

[25] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 53(12):4655–4666, 2007.

[26] L. Wang, G. Chen, and H. Li. Group scad regression analysis for microarray time course gene expression data. *Bioinformatics*, 23(12):1486–1494, 2007.

[27] S. J. Wright, R. D. Nowak, and M. A. Figueiredo. Sparse reconstruction by separable approximation. *Signal Processing, IEEE Transactions on*, 57(7):2479–2493, 2009.

[28] S. Xiang, X. Shen, and J. Ye. Efficient Sparse Group Feature Selection via Nonconvex Optimization. In *The 30th International Conference on Machine Learning (ICML)*, pages 284–292, 2013.

[29] S. Xiang, L. Yuan, W. Fan, Y. Wang, P. M. Thompson, and J. Ye. Multi-source learning with block-wise missing data for alzheimer's disease prediction. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 185–193. ACM, 2013.

[30] Y. Xu and D. Rockmore. Feature selection for link prediction. In *Proceedings of the 5th Ph. D. workshop on Information and knowledge*, pages 25–32. ACM, 2012.

[31] J. Ye and J. Liu. Sparse methods for biomedical data. *ACM SIGKDD Explorations Newsletter*, 14(1):4–15, 2012.

[32] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[33] T. Zhang. Multi-stage convex relaxation for feature selection. *Bernoulli*, 19(5B):2277–2293, 2011.

[34] H. Zou and R. Li. One-step sparse estimates in nonconcave penalized likelihood models. *Annals of statistics*, 36(4):1509, 2008.

APPENDIX

A. PROOF OF THEOREM 1

PROOF. Let $\boldsymbol{w}^k$ denote $\boldsymbol{x}^k - \nabla f(\boldsymbol{x}^k)$. It is clear that

$$\begin{aligned}
&\|\boldsymbol{x}^{k+1} - \boldsymbol{w}^k\|_2^2 \\
=&\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\|_2^2 + \|\boldsymbol{x}^* - \boldsymbol{w}^k\|_2^2 + 2\langle \boldsymbol{x}^{k+1} - \boldsymbol{x}^*, \boldsymbol{x}^* - \boldsymbol{w}^k\rangle \\
\leq&\|\boldsymbol{x}^* - \boldsymbol{w}^k\|_2^2,
\end{aligned}$$

where the last inequality comes from the optimality of $\boldsymbol{x}^{k+1}$. After eliminating $\|\boldsymbol{x}^* - \boldsymbol{w}^k\|_2^2$ from both sides we can obtain:

$$\begin{aligned}
&\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\|_2 \\
\leq&2\langle \boldsymbol{x}^{k+1} - \boldsymbol{x}^*, \frac{\boldsymbol{w}^k - \boldsymbol{x}^*}{\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\|_2}\rangle \\
=&2\langle \boldsymbol{x}^k - \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x}^k - \boldsymbol{y}) - \boldsymbol{x}^*, \frac{\boldsymbol{x}^{k+1} - \boldsymbol{x}^*}{\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\|_2}\rangle \\
=&2\langle \boldsymbol{x}^k - \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x}^k - (\boldsymbol{A}\boldsymbol{x}^* + \boldsymbol{e}^*)) - \boldsymbol{x}^*, \frac{\boldsymbol{x}^{k+1} - \boldsymbol{x}^*}{\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\|_2}\rangle \\
=&2\langle (\boldsymbol{I} - \boldsymbol{A}^T\boldsymbol{A})(\boldsymbol{x}^k - \boldsymbol{x}^*) - \boldsymbol{A}^T\boldsymbol{e}^*, \frac{\boldsymbol{x}^{k+1} - \boldsymbol{x}^*}{\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\|_2}\rangle \\
=&2\langle (\boldsymbol{I} - \boldsymbol{A}_U^T\boldsymbol{A}_U)(\boldsymbol{x}^k - \boldsymbol{x}^*) - \boldsymbol{A}^T\boldsymbol{e}^*, \frac{\boldsymbol{x}^{k+1} - \boldsymbol{x}^*}{\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\|_2}\rangle \\
\leq&2(\|\boldsymbol{I} - \boldsymbol{A}_U^T\boldsymbol{A}_U\|_2\|\boldsymbol{x}^k - \boldsymbol{x}^*\|_2 + \|\boldsymbol{A}\frac{\boldsymbol{x}^{k+1} - \boldsymbol{x}^*}{\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\|_2}\|_2\|\boldsymbol{e}^*\|_2) \\
\leq&2(c_1\|\boldsymbol{x}^k - \boldsymbol{x}^*\|_2 + \sqrt{1+c_2}\|\boldsymbol{e}^*\|_2),
\end{aligned}$$

where the set U is the union of support of $\boldsymbol{x}^*$, $\boldsymbol{x}^k$ and $\boldsymbol{x}^{k+1}$ and the last inequality is from the fact that the spectral norm of $I - A_U^T A_U$ is upperbounded by $\delta_{|U|}$ [6]. The first conclusion then follows from expanding the last term and compute the power series.

To prove the second result, a finer treatment of the set U above is needed. Specifically, we consider the following four sets:

$$\begin{aligned}
&I_1 = \text{supp}(\boldsymbol{x}^k), \quad I_2 = \text{supp}(\boldsymbol{x}^{k+1}) \\
&I_3 = \text{supp}(\boldsymbol{x}^*) - \text{supp}(\boldsymbol{x}^k) \\
&I_4 = \text{supp}(\boldsymbol{x}^*) - \text{supp}(\boldsymbol{x}^{k+1}),
\end{aligned}$$

and it is easy to veryfy that:

$$\begin{aligned}
&\text{supp}(\boldsymbol{x}^k - \boldsymbol{x}^*) \subset I_{13} \\
&\text{supp}(\boldsymbol{x}^{k+1} - \boldsymbol{x}^*) \subset I_{24} \\
&|I_{ij}| = |I_i \cup I_j| \leq 2s_1, \quad \forall (i,j) \in \{1,2,3,4\}.
\end{aligned}$$

Therefore we can conclude that:

$$\begin{aligned}
&(\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\|_2 - 2\sqrt{1+c_2}\|\boldsymbol{e}^*\|_2)\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\|_2 \\
\leq&2\langle (\boldsymbol{I} - \boldsymbol{A}^T\boldsymbol{A})(\boldsymbol{x}^k - \boldsymbol{x}^*), \boldsymbol{x}^{k+1} - \boldsymbol{x}^*\rangle \\
=&2\langle (\boldsymbol{I} - \boldsymbol{A}^T\boldsymbol{A})\left((\boldsymbol{x}^k - \boldsymbol{x}^*)_{I_1} + (\boldsymbol{x}^k - \boldsymbol{x}^*)_{I_3}\right), \\
&(\boldsymbol{x}^{k+1} - \boldsymbol{x}^*)_{I_2} + (\boldsymbol{x}^{k+1} - \boldsymbol{x}^*)_{I_4}\rangle \\
\leq&2\langle (\boldsymbol{I} - \boldsymbol{A}_{I_{12}}^T\boldsymbol{A}_{I_{12}})(\boldsymbol{x}^k - \boldsymbol{x}^*)_{I_1}, (\boldsymbol{x}^{k+1} - \boldsymbol{x}^*)_{I_2}\rangle \\
&+ 2\langle (\boldsymbol{I} - \boldsymbol{A}_{I_{14}}^T\boldsymbol{A}_{I_{14}})(\boldsymbol{x}^k - \boldsymbol{x}^*)_{I_1}, (\boldsymbol{x}^{k+1} - \boldsymbol{x}^*)_{I_4}\rangle \\
&+ 2\langle (\boldsymbol{I} - \boldsymbol{A}_{I_{32}}^T\boldsymbol{A}_{I_{32}})(\boldsymbol{x}^k - \boldsymbol{x}^*)_{I_3}, (\boldsymbol{x}^{k+1} - \boldsymbol{x}^*)_{I_2}\rangle \\
&+ 2\langle (\boldsymbol{I} - \boldsymbol{A}_{I_{34}}^T\boldsymbol{A}_{I_{34}})(\boldsymbol{x}^k - \boldsymbol{x}^*)_{I_3}, (\boldsymbol{x}^{k+1} - \boldsymbol{x}^*)_{I_4}\rangle \\
\leq&2c_2(\|(\boldsymbol{x}^k - \boldsymbol{x}^*)_{I_1}\|_2\|(\boldsymbol{x}^{k+1} - \boldsymbol{x}^*)_{I_2}\|_2 \\
&+ \|(\boldsymbol{x}^k - \boldsymbol{x}^*)_{I_1}\|_2\|(\boldsymbol{x}^{k+1} - \boldsymbol{x}^*)_{I_4}\|_2 \\
&+ \|(\boldsymbol{x}^k - \boldsymbol{x}^*)_{I_3}\|_2\|(\boldsymbol{x}^{k+1} - \boldsymbol{x}^*)_{I_2}\|_2 \\
&+ \|(\boldsymbol{x}^k - \boldsymbol{x}^*)_{I_3}\|_2\|(\boldsymbol{x}^{k+1} - \boldsymbol{x}^*)_{I_4}\|_2) \\
\leq&2c_2\sqrt{2\|(\boldsymbol{x}^k - \boldsymbol{x}^*)_{I_1}\|_2^2 + 2\|(\boldsymbol{x}^k - \boldsymbol{x}^*)_{I_3}\|_2^2} \\
&\sqrt{2\|(\boldsymbol{x}^{k+1} - \boldsymbol{x}^*)_{I_2}\|_2^2 + 2\|(\boldsymbol{x}^{k+1} - \boldsymbol{x}^*)_{I_4}\|_2^2} \\
=&4c_2\|\boldsymbol{x}^k - \boldsymbol{x}^*\|_2\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^*\|_2,
\end{aligned}$$

where the first inequality is from our proof of the first result and we apply the Cauchy inequality to obtain the last inequality. The proof is completed by expanding the last term and computing the resulting power series. □