Absolute Fused Lasso and Its Application to Genome-Wide Association Studies

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

Jun Liu
SAS Institute Inc.
Cary, NC 27513
jun.liu@sas.com

Pinghua Gong
University of Michigan
Ann Arbor, MI 48109
gongp@umich.edu

Ruiwen Zhang
SAS Institute Inc.
Cary, NC 27513
ruiwen.zhang@sas.com

Xiaotong Shen
University of Minnesota
Minneapolis, MN 55455
xshen@stat.umn.edu

Jieping Ye
University of Michigan
Ann Arbor, MI 48109
jpye@umich.edu

ABSTRACT

In many real-world applications, the samples/features acquired are in spatial or temporal order. In such cases, the magnitudes of adjacent samples/features are typically close to each other. Meanwhile, in the high-dimensional scenario, identifying the most relevant samples/features is also desired. In this paper, we consider a regularized model which can simultaneously identify important features and group similar features together. The model is based on a penalty called Absolute Fused Lasso (AFL). The AFL penalty encourages sparsity in the coefficients as well as their successive differences of absolute values—i.e., local constancy of the coefficient components in absolute values. Due to the non-convexity of AFL, it is challenging to develop efficient algorithms to solve the optimization problem. To this end, we employ the Difference of Convex functions (DC) programming to optimize the proposed non-convex problem. At each DC iteration, we adopt the proximal algorithm to solve a convex regularized sub-problem. One of the major contributions of this paper is to develop a highly efficient algorithm to compute the proximal operator. Empirical studies on both synthetic and real-world data sets from Genome-Wide Association Studies demonstrate the efficiency and effectiveness of the proposed approach in simultaneous identifying important features and grouping similar features.

Keywords

Absolute Fused Lasso; Non-convex Optimization; Proximal Operator; GWAS

1. INTRODUCTION

Regularized learning methods have recently attracted increasing attention in various applications. A common scenario that occurs in many studies is that the data sets we investigated are of some natural (e.g., spatial or temporal) order; examples include the comparative genomic hybridization data [18], prostate cancer data [17] and neuroimaging data [23]. For those classes of studies, it is often the case that the adjacent samples/features are similar and even identical. Moreover, in Genome-Wide Association Studies (GWAS), a causal single-nucleotide polymorphism (SNP) often exhibits high similarity with its nearby SNPs. It is thus desired to group nearby SNPs together. However, due to the ambiguity choice of reference allele during genotype coding [8], we should group adjacent SNPs if their absolute values are close to each other.

Previous works [22, 26, 1, 25, 20] indicate that utilizing the inherent structure information in the data can potentially be beneficial for model construction and interpretation. Thus if the data exhibits some sequential order, we can potentially incorporate such prior knowledge into the model to improve performance. Meanwhile, due to the curse of dimensionality in the high-dimensional scenario, identifying the most relevant features is of crucial importance. In such a case, the traditional Lasso [16] model is insufficient to produce desired results since it tends to select only one of those highly correlated features [29]. There are mainly two approaches in the literature to address the problem. One approach adopts the fused penalty (e.g., fused Lasso), which can yield a sparse solution in both the coefficients and their successive differences [17, 18, 9]. However, it does not consider the case that adjacent features have high similarity but opposite signs. Another approach utilizes the graph structure among features (e.g., OSCAR) during model construction [3, 23, 28]. However, such an approach is too general and does not make full use of the specific structure of the genome data.

Generally, a GWA study focuses on investigating associations between genotypes (SNPs) and disease phenotypes. Previous studies [8, 28] have shown that incorporating the linkage disequilibrium (LD) information [13] between adjacent SNPs is beneficial in delineating association SNPs with smoothness and less randomness than individual SNP analysis. The studies in [8] also argue that the fused Lasso is not effective due to the ambiguity choice of coding reference. Thus, it is desired to penalize successive SNPs whose absolute values are close or identical.

In this paper, we consider a regularized model which uses a penalty called "Absolute Fused Lasso" (AFL) to solve such a problem. The AFL penalty encourages sparsity in the co-

KDD '16, August 13-17, 2016, San Francisco, CA, USA

DOI: http://dx.doi.org/10.1145/2939672.2939827

efficients as well as their successive differences of absolute values—i.e., local constancy of the coefficient components in absolute value. With AFL, highly similar features can potentially be grouped together even when their signs are different. Since the AFL problems are non-convex, it is challenging to develop efficient optimization algorithms. To this end, we employ the Difference of Convex functions (DC) programming to solve the non-convex problem. At each DC iteration, we adopt the proximal algorithm to efficiently solve the convex subproblem, which iteratively solves a proximal operator problem; we further use the Barzilai-Borwein (BB) rule for line search to accelerate convergence. One of the major contributions of this paper is to show that the proximal operator problem can be solved efficiently. Specifically, by exploiting the special structure of the regularizer, we first convert the computation of such proximal operator to an equivalent optimization problem via a Euclidean projection onto a special polyhedron. We then develop a gradient descent algorithm based on a novel restart technique by utilizing the optimality condition to efficiently solve the projection problem. We have conducted empirical evaluations on both synthetic data and real data. Experimental results demonstrate that the proposed DC-Proximal approach can achieve up to 50x speedup over general DC-ADMM (alternating direction method of multipliers) method—it allows us to perform efficient AFL modeling on large-scale genome data that contains tens of thousands SNPs.

2. THE AFL FORMULATION

In this paper, we consider the following Absolute Fused Lasso (AFL) regularization model:

$$\min_{\mathbf{x}\in\mathbb{R}^p} \text{loss}(\mathbf{x}) + \text{afl}(\mathbf{x}), \tag{1}$$

where $\text{loss}(\mathbf{x})$ is a convex empirical loss function (e.g., the least squares loss or the logistic loss) and the AFL penalty is defined as:

$$\text{afl}(\mathbf{x}) = \lambda_1\|\mathbf{x}\|_1 + \lambda_2\sum_{i=1}^{p-1}||x_i| - |x_{i+1}||, \tag{2}$$

where λ_1 and λ_2 are non-negative regularization parameters. The second term penalizes differences of successive coefficients' magnitudes and can be considered as a grouping penalty. By imposing both the l_1 and the grouping penalties, the AFL model can simultaneously identify important features as well as group similar (identical) features together.

Different from the fused Lasso that penalizes the l_1-norm on successive differences of coefficients (i.e., $\lambda_2\sum_{i=1}^{p-1}|x_i - x_{i+1}|$), the AFL encourages the smoothness of adjacent coefficients whose absolute values are close or even identical. Thus, strong successive signals can be identified by Eq. (1) even when their signs are different. In general, adopting the AFL penalty is expected to be more effective than the fused Lasso (See an example in Fig. 1). Note that in GWAS, the SNPs data we obtain through genotype coding are strongly affected by the choice of reference allele. Thus it is insufficient to just penalize the successive differences without considering the absolute values. In [8], the authors use the l_2-norm on the absolute difference, and apply coordinate descent to solve the proposed formulation. However, due to the use of l_2-norm, the fused property, i.e., the absolute values of nearby terms tend to be identical, does not hold any more.

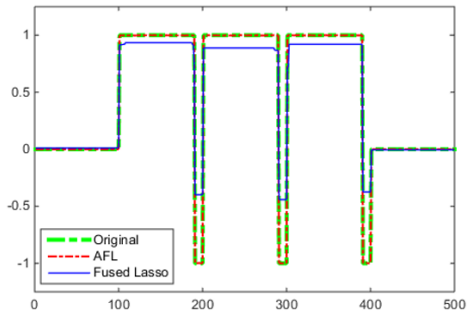


Figure 1: Comparison of the solutions of the AFL and the fused Lasso on a simulated data. The AFL (red line) provides better recovery of the original signals (green) than the fused Lasso (blue). See Supplement A for more details about this experiment.

In this paper, we propose to adopt the DC programming to solve the AFL problem (1) and apply the proximal algorithm to solve the sub-problem at each DC iteration. One of our main technical contributions is to develop an algorithm to efficiently solve the proximal operator problem by exploiting the special structure of the regularizer, which is a key building block of the proximal algorithm.

3. DC PROGRAMMING FOR SOLVING THE AFL PROBLEM

The AFL formulation in Eq. (1) is a non-convex optimization problem. We propose to use the Difference of Convex functions (DC) programming [15, 14] to solve it, where a key step is to decompose the objective function in Eq. (1) into the difference of two convex functions. By noting that

$$||x_i| - |x_{i+1}|| = |x_i + x_{i+1}| + |x_i - x_{i+1}| - (|x_i| + |x_{i+1}|),$$

we decompose the objective function in Eq. (1) into the difference of the following two functions:

$$f_1(\mathbf{x}) = \text{loss}(\mathbf{x}) + \lambda_1\|\mathbf{x}\|_1 + \lambda_2\sum_{i=1}^{p-1}(|x_i + x_{i+1}| + |x_i - x_{i+1}|),$$

$$f_2(\mathbf{x}) = \lambda_2\sum_{i=1}^{p-1}(|x_i| + |x_{i+1}|).$$

By linearization of $f_2(\mathbf{x})$, the per-iteration subproblem of the DC algorithm can be written as:

$$\min_{\mathbf{x}} \text{loss}(\mathbf{x}) - (\mathbf{c}^k)^T\mathbf{x} + \lambda_1\|\mathbf{x}\|_1 + 2\lambda_2\sum_{i=1}^{p-1}\max(|x_i|, |x_{i+1}|), \tag{3}$$

where

$$c_i^k = \lambda_2 d_i\,\text{sgn}(x_i^k) \text{ with } d_1 = d_p = 1, d_i = 2, 2 \le i \le p-1 \tag{4}$$

and $\text{sgn}(\cdot)$ is the signum function (Detailed derivation is provided in Supplement B). We summarize the DC programming in Algorithm 1. A key building block in this algorithm is how to efficiently solve the subproblem (3). Next, we show that (3) can be efficiently solved via a proximal algorithm.

Algorithm 1 DC algorithm for solving the AFL Problem

Input: data matrix $\mathbf{A} \in \mathbb{R}^{n\times p}$, response vector $\mathbf{y} \in \mathbb{R}^{n\times 1}$, regularizes λ_1, λ_2, and tolerance ϵ
Output: $\mathbf{x}$

Initialization: $\mathbf{x}^0 \leftarrow \mathbf{0}, k = 0$
while $f(\mathbf{x}^k) - f(\mathbf{x}^{k+1}) > \epsilon$ **do**
  Update $\mathbf{c}^k$ according to Eq. (4).
  Update $\mathbf{x}^{k+1}$ according to Eq. (3).
  $k \leftarrow k + 1$.
end while

4. THE PROXIMAL ALGORITHM

In this paper, we adopt the proximal framework [21] to solve the sub-optimization problem (3) at each DC iteration. Problem (3) is equivalent to

$$\min_{\mathbf{x}\in\mathbb{R}^p} h(\mathbf{x}) = l(\mathbf{x}) + m(\mathbf{x}), \tag{5}$$

where

$$l(\mathbf{x}) = \text{loss}(\mathbf{x}) - (\mathbf{c}^k)^T\mathbf{x},$$
$$m(\mathbf{x}) = \lambda_1\|\mathbf{x}\|_1 + 2\lambda_2\sum_{i=1}^{p-1}\max(|x_i|, |x_{i+1}|).$$

The proximal algorithm solves problem (3) by generating a sequence $\{\mathbf{x}^k\}$ by solving:

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}\in\mathbb{R}^p}\{l(\mathbf{x}^k) + \langle\nabla l(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k\rangle + m(\mathbf{x}) + \frac{t^k}{2}\|\mathbf{x} - \mathbf{x}^k\|_2^2\}, \tag{6}$$

where $t^k > 0$ is chosen by some rule introduced below. It is easy to show that (6) is equivalent to the following proximal operator problem:

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}\in\mathbb{R}^p}\frac{1}{2}\|\mathbf{x} - \mathbf{u}^k\|^2 + \frac{1}{t^k}m(\mathbf{x}), \tag{7}$$

where $\mathbf{u}^k = \mathbf{x}^k - \nabla l(\mathbf{x}^k)/t^k$. Thus, it can be viewed as the gradient descent along the direction $-\nabla l(\mathbf{x}^k)$ with the step size $1/t^k$ plus computing the proximal operator problem (7). The pseudo codes of the algorithm are summarized in Algorithm 2.

Algorithm 2 The Proximal Algorithm

Input: $\mathbf{A}, \mathbf{y}, \lambda_1, \lambda_2$
Output: $\mathbf{x}$

Choose $\eta > 1$, $t_{max} > t_{min} > 0$
Initialization: $\mathbf{x}^0, k = 0$
while some stopping criterion is not satisfied **do**
  Choose $t^k \in [t_{min}, t_{max}]$
  while line search criterion is not satisfied **do**
    Update $\mathbf{x}^{k+1}$ according to Eq. (7).
    $t^k \leftarrow \eta t^k$.
  end while
  $k \leftarrow k + 1$.
end while

To guarantee convergence, we use a line search criterion to choose an appropriate step size. Specifically, we accept the step size $1/t^k$ if the following inequality holds:

$$h(\mathbf{x}^{k+1}) \le h(\mathbf{x}^k) - \frac{\sigma}{2}t^k\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2,$$

where $\sigma \in (0, 1)$ is a constant. To further accelerate the convergence speed of the proximal algorithm, as suggested by the studies in [21, 5], we adopt the Barzilai-Borwein (BB) rule to initialize the line search step size as $1/t^{k,0}$, where

$$t^{k,0} = \frac{\langle\mathbf{a}^k, \mathbf{b}^k\rangle}{\langle\mathbf{a}^k, \mathbf{a}^k\rangle}$$

with $\mathbf{a}^k = \mathbf{x}^k - \mathbf{x}^{k-1}$ and $\mathbf{b}^k = \nabla l(\mathbf{x}^k) - \nabla l(\mathbf{x}^{k-1})$.

Notice that a key step in the proximal algorithm is how to efficiently solve the proximal operator problem (7). In the next section, we introduce our efficient approach to solve (7) by exploiting the special structure of the regularizer.

5. EFFICIENT COMPUTATION OF THE PROXIMAL OPERATOR

For discussion convenience, we absorb t^k into the regularization parameters λ_1 and λ_2, and omit the superscript k in Eq. (7). Then the proximal operator problem (7) can be simplified as follows:

$$\pi_{\lambda_2}^{\lambda_1}(\mathbf{u}) = \arg\min_{\mathbf{x}\in\mathbb{R}^p}\{\frac{1}{2}\|\mathbf{x} - \mathbf{u}\|^2 + \lambda_1\|\mathbf{x}\|_1 + 2\lambda_2\sum_{i=1}^{p-1}\max(|x_i|, |x_{i+1}|)\}. \tag{8}$$

By applying the procedure discussed in [4], we have the following theorem:

Theorem 1. *For any $\lambda_1, \lambda_2 \ge 0$, we have*

$$\pi_{\lambda_2}^{\lambda_1}(\mathbf{u}) = \text{sgn}(\pi_{\lambda_2}^0(\mathbf{u})) \odot \max(|\pi_{\lambda_2}^0(\mathbf{u})| - \lambda_1, 0). \tag{9}$$

Theorem 1 implies that we can solve problem (8) in two steps: first solve (8) with $\lambda_1 = 0$ and then applying (9) to obtain the final result. Let $\lambda = 2\lambda_2$ and $\lambda_1 = 0$, Eq. (8) can be rewritten as:

$$\pi_\lambda(\mathbf{u}) = \arg\min_{\mathbf{x}\in\mathbb{R}^p}\{\frac{1}{2}\|\mathbf{x} - \mathbf{u}\|^2 + \lambda\sum_{i=1}^{p-1}\max(|x_i|, |x_{i+1}|)\}. \tag{10}$$

We propose to solve problem (10) efficiently by converting the proximal operator to a Euclidean projection onto a special polyhedron. To perform this transformation, we utilize some important properties of (10) as summarized in Lemma 1, where a detailed proof is provided in Supplement C.

Lemma 1. *Let $\mathbf{x}^* = \pi_\lambda(\mathbf{u})$ be the optimal solution to* (10). *$\forall\lambda > 0$, we have:*

i) if $u_i \ge 0$, then $u_i \ge x_i^* \ge 0$,
ii) if $u_i < 0$, then $u_i \le x_i^* \le 0$,
iii) $\pi_\lambda(\mathbf{u}) = \text{sgn}(\mathbf{u}) \odot \pi_\lambda(|\mathbf{u}|)$,
iv) if $|u_i| \ge |u_{i+1}|$, then $|x_i^*| \ge |x_{i+1}^*|$,
v) if $|u_i| < |u_{i+1}|$, then $|x_i^*| \le |x_{i+1}^*|$.

5.1 Equivalent Euclidean Projection Problem

Assume $\mathbf{u} \ge 0$, we define a sparse matrix $R \in \mathbb{R}^{(p-1)\times p}$ as follows:

$$R_{ij} = \begin{cases} 1 & u_i < u_{i+1}, j = i \\ 1 & u_i \ge u_{i+1}, j = i+1 \\ -1 & u_i \ge u_{i+1}, j = i \\ -1 & u_i < u_{i+1}, j = i+1 \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

In addition, we denote a vector $\mathbf{w} \in \mathbb{R}^p$ with the j-th entry defined as:

$$w_j = \begin{cases} 2 & \sum_i R_{ij} = 2 \\ 0 & \sum_i R_{ij} \leq -1 \\ 1 & \text{otherwise.} \end{cases} \quad (12)$$

With Lemma 1 and the above definitions of R and w, we next present the following theorem which converts the proximal operator problem to an equivalent Euclidean projection problem.

Theorem 2. *Let* $\mathbf{u} \geq 0$ *and* $\lambda > 0$. *Let*

$$\mathbf{v} = \mathbf{u} - \lambda \mathbf{w} \quad (13)$$

and

$$P = \{\mathbf{x} | R\mathbf{x} \leq 0, \mathbf{x} \geq 0\}. \quad (14)$$

Define the Euclidean projection of $\mathbf{v}$ *onto* P *as:*

$$\pi_\lambda^P(\mathbf{v}) = \arg\min_{\mathbf{x} \in P} \frac{1}{2}\|\mathbf{x} - \mathbf{v}\|^2. \quad (15)$$

We have

$$\pi_\lambda(\mathbf{u}) = \pi_\lambda^P(\mathbf{v}). \quad (16)$$

The above theorem implies that, the proximal operator in (10) can be solved by solving the Euclidean projection problem in (15). To further simplify, our next theorem shows that, such a Euclidean projection problem can be solved by a simplified problem without the non-negative constraint.

Theorem 3. *Let* $\mathbf{u} \geq 0$, $\lambda > 0$,

$$Q = \{\mathbf{x} | R\mathbf{x} \leq 0\}, \quad (17)$$

and

$$\pi_\lambda^Q(\mathbf{v}) = \arg\min_{\mathbf{x} \in Q} \frac{1}{2}\|\mathbf{x} - \mathbf{v}\|^2. \quad (18)$$

We have

$$\pi_\lambda^P(\mathbf{v}) = \max(\pi_\lambda^Q(\mathbf{v}), 0). \quad (19)$$

Detailed proofs of Theorem 2 and Theorem 3 are provided in Supplements D & E. In the next section, we discuss a restart technique to efficiently solve the Euclidean projection problem (18).

5.2 The Restart Technique

Introducing the dual variable $\mathbf{z} \in \mathbb{R}^{p-1}$ for the inequality constraints in (18), we can obtain the Lagrangian in Supp. (E-48). The dual problem of (18) is equivalent to

$$\min_{\mathbf{z} \geq 0} \{\phi(\mathbf{z}) = \frac{1}{2}\|R^T\mathbf{z} - \mathbf{v}\|^2\}. \quad (20)$$

We propose to solve (18) by simultaneously using the information of the primal and dual problems. The novelty lies in the usage of the so-called restart technique for fast convergence.

5.2.1 Optimality Condition and the Support Set

Our proposed restart technique is built on the introduction of the *support set*. Specifically, $\forall \mathbf{z} \geq 0$ and denote $\mathbf{g} = \phi'(\mathbf{z})$, we define the support set as follows:

$$S(\mathbf{z}) = \{i : i \in [1, p-1], z_i = 0, g_i > 0\} \cup \{0, p\}. \quad (21)$$

The support set $S(\mathbf{z})$ is motivated by the optimality of the problem (20), and shall be used for defining a nonlinear and discontinuous mapping from $\mathbf{z}$ to $\mathbf{x}$. $\forall \mathbf{z}^* \geq 0$, it is a minimizer of (20) if and only if $\langle \mathbf{z} - \mathbf{z}^*, \phi'(\mathbf{z}^*) \rangle \geq 0, \forall \mathbf{z} \geq 0$.

From the optimality condition, we can build the relationship between the minimizer and its gradient, as summarized in the following lemma:

Lemma 2. *Let* $\mathbf{z}^*$ *be the optimal solution to* (20) *and* $\mathbf{g}^* = \phi'(\mathbf{z}^*)$. *We have: i) if* $z_i^* > 0$ *then* $g_i^* = 0$, *and ii) if* $g_i^* > 0$, *then* $z_i^* = 0$.

The matrix RR^T is very special, and it can be shown that its eigenvalues are $2 - 2\cos(i\pi/p), i = 1, 2, \ldots, p-1$, and thus it is positive definite. Note that RR^T is the Hessian of $\phi(\mathbf{z})$, which implies that the minimizer of (20) is unique.

5.2.2 A Nonlinear Mapping $\omega(\cdot)$ from $\mathbf{z}$ to $\mathbf{x}$

Let $s_0 = 0$ denote the smallest entry in $S(\mathbf{z})$, and $s_{|S|} = p$ denote the largest entry in $S(\mathbf{z})$. In addition, we denote the j-th largest entry in the set $S - \{0, p\}$ by $s_j, j = 1, 2, \ldots, |S| - 2$. It is clear that $1 \leq s_1$ and $s_{|S|-2} \leq p - 1$. With $s_0, s_1, \ldots, s_{|S|-1}$, the indices in $[1 : p]$ can be divided into $|S| - 1$ non-overlapping groups:

$$G_j = \{i : s_{j-1} + 1 \leq i \leq s_j\}, 1 \leq j \leq |S| - 1. \quad (22)$$

Let $\mathbf{e} \in \mathbf{R}^p$ be a vector composed of 1's, and $\mathbf{e}_{G_j}$ and $\mathbf{v}_{G_j}$ be the j-th group of $\mathbf{e}$ and $\mathbf{v}$ corresponding to the indices in G_j, respectively. For discussion convenience, assume $z_0 = z_p = 0$, then we can define the nonlinear mapping $\mathbf{x} = \omega(\mathbf{z})$ based on the support set S as:

$$x_i = \frac{\langle \mathbf{e}_{G_j}, \mathbf{v}_{G_j} \rangle - z_{s_{j-1}} + z_{s_j}}{|G_j|}, i \in G_j, 1 \leq j \leq |S| - 1. \quad (23)$$

With Lemma 2 and the definition of support set in (21), it is easy to show that the optimal solution to problem (18) can be exactly recovered by the support set $S(\mathbf{z}^*)$, as stated in the following theorem.

Theorem 4. *Let* $\mathbf{z}^*$ *be the minimizer of the dual problem* (20), *and* $\mathbf{x}^*$ *be the minimizer of primal problem* (18). *Then* $\mathbf{x}^*$ *can be recovered by* $\mathbf{x}^* = \omega(\mathbf{z}^*)$.

5.2.3 The Restart Technique and Properties

By introducing the support set S, Theorem 4 provides an alternative efficient way to computing $\mathbf{x}^*$ from $\mathbf{z}^*$. Specifically, we can exactly obtain $\mathbf{x}^* = \omega(\tilde{\mathbf{z}})$, where $\tilde{\mathbf{z}}$ is an appropriate solution with $S(\tilde{\mathbf{z}}) = S(\mathbf{z}^*)$ even if $\tilde{\mathbf{z}} \neq \mathbf{z}^*$. The intuition is that, for a given appropriate solution $\tilde{\mathbf{z}} \neq \mathbf{z}^*$, if $S(\tilde{\mathbf{z}})$ is close to $S(\mathbf{z}^*)$, $\mathbf{x} = \omega(\tilde{\mathbf{z}})$ can be a better approximation than $\tilde{\mathbf{x}} = \mathbf{v} - R^T\tilde{\mathbf{z}}$ for the primal.

We then present a gradient projection algorithm based on the proposed restart technique, as summarized in Algorithm 3. Given an iterative solution $\mathbf{z}^k$, we do not perform the gradient projection at the point $\mathbf{z} = \mathbf{z}^k$. Instead, we first compute $\mathbf{x}^k = \omega(\mathbf{z}^k)$. Then, we compute a restart point $\mathbf{z}_0^k$ by $\mathbf{x}^k = \mathbf{v} - R^T\mathbf{z}_0^k$, where $\mathbf{z}_0^k$ can be solved by an equivalent linear system $RR^T\mathbf{z}_0^k = R\mathbf{v} - R\mathbf{x}^k$. Finally, we perform the gradient projection at the restart point $\mathbf{z} = \mathbf{z}_0^k$. Note that $P_0(\mathbf{x})$ is an operator that projects $\mathbf{x}$ onto the non-negative orthant.

Algorithm 3 Gradient Projection Algorithm with a Restart Technique

Input: $\mathbf{v}, \lambda, R$
Output: $\mathbf{z}$

1: Initialization: $\mathbf{z}^0 \leftarrow \mathbf{0}, L = 2 - 2\cos(\pi(p-1)/p)$, $k = 0$;
2: Compute $\mathbf{g}^0 = \phi'(\mathbf{z}^0) = RR^T\mathbf{z}^0 - R\mathbf{v}$;
and set $\mathbf{z}^0 = P_0(\mathbf{z}^0 - \mathbf{g}^0/L)$;
3: **while** not converge **do**
4: Update the support set $S(\mathbf{z}^k)$ according to (21);
5: Update $\mathbf{x}^k = \omega(\mathbf{z}^k)$ according to (23);
6: Compute $\mathbf{z}_0^k$ as the solution to $RR^T\mathbf{z}_0^k = R\mathbf{v} - R\mathbf{x}^k$;
7: Update $\mathbf{z}^{k+1} = P_0(\mathbf{z}_0^k)$, and set $k \leftarrow k+1$;
8: **end while**

5.3 Discussion

To end this section, we summarize our methodology for solving the proximal operator problem (8) as follows. We first show that a minimizer of problem (8) can be obtained by applying a soft-thresholding (9) on the solution of an alternative optimization problem (10). By applying the properties of (10) introduced in Lemma 1 and two variables R and w defined in (11) and (12), we show that the proximal operator problem (10) can be convert to an equivalent problem (15). In the sequel, we present to optimize an alternative problem (18) without the non-negative constraint through (19). To solve problem (18), we develop a novel restart technique by introducing the support set (21) and a nonlinear mapping (22). We propose to use Algorithm 3 to solve (18) for efficient computation.

6. EXPERIMENTS

In this section, we evaluate the AFL model (with the least-squares loss) and the proposed algorithm on both synthetic and real-world data. We first evaluate the efficiency of our proposed algorithm in §6.1.1, and then compare the AFL with the fused Lasso in §6.1.2. Next, we evaluate the prediction performance of the AFL in two GWA studies in §6.2.1 and §6.2.2. Finally, we show the effectiveness of the AFL in identifying genetic risk factors in §6.2.2. For all experiments, we use the following two stopping criteria for our algorithm: 1) the relative difference of function values between two iterations is less than a tolerance of 10^{-5}, and 2) the algorithm exceeds the maximum iterations (1000 iterations).

6.1 Evaluation on Synthetic Data

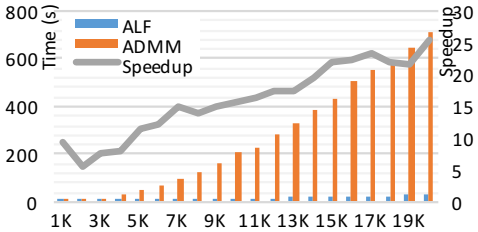
6.1.1 *Efficiency of AFL*

We present the empirical studies on the efficiency of our proposed algorithm by comparing our method with the approach that adopts the alternating direction method of multipliers (ADMM) to solve the sub-problem at each DC iteration. The experiments are carried out on a collection of randomly generated data sets $\mathbf{A} \in \mathbb{R}^{n\times p}$ and outcomes $\mathbf{y} \in \mathbb{R}^{n\times 1}$. In addition, denote $\bar{\lambda} = \|\mathbf{A}^T\mathbf{y}\|_\infty$. We then conduct the evaluations in the following two scenarios:

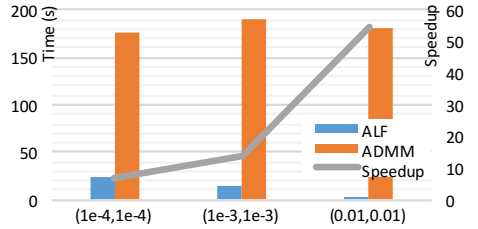
Scenario 1. Varying the number of features p with a fixed sample size and fixed regularization parameters λ_1 and λ_2. We fix the number of samples $n = 500$ and vary the number of features p from 1,000 to 20,000. We set the regularizers as $\lambda_1 = \lambda_2 = 10^{-3}\bar{\lambda}$.

Scenario 2. Varying regularization parameters λ_1 and λ_2 with a fixed sample/feature size. We fix the $n = 500$ and $p = 10,000$. We choose the values of (λ_1, λ_2) from the following set: $\{(10^{-4}\bar{\lambda}, 10^{-4}\bar{\lambda}), (10^{-3}\bar{\lambda}, 10^{-3}\bar{\lambda}), (0.01\bar{\lambda}, 0.01\bar{\lambda})\}$.

Figure 2 summarizes the running time (in seconds) and speedup of AFL (proximal algorithm) over ADMM in the above two scenarios. From these figures, we have the following observations: (1) Our proposed algorithm is much more efficient than ADMM in both scenarios. (2) The speedup of AFL over ADMM increases as the feature size increases. This indicates that our proposed approach using DC programming and the proximal algorithm is capable of handling large-scale learning problems. (3) The speedup of AFL over ADMM increases as the regularized parameters become larger. Thus, our method is expected to be superior over ADMM in real-world applications, i.e., only a small number of features are relevant—a relatively large regularized parameter value is preferred.



(a) Scenario 1



(b) Scenario 2

Figure 2: Comparison of running times and speedups of AFL over ADMM.

6.1.2 *Comparison of AFL and Fused Lasso*

In this study, we compare the AFL model with the fused Lasso. Recall that the AFL is designed to encourage the smoothness of adjacent coefficients whose absolute values are close or even identical. Thus if the adjacent features exhibit different signs in the model, the AFL approach is expected to be more effective than the fused Lasso.

We generate the synthetic data via the linear model $\mathbf{y} = \mathbf{A}\bar{\mathbf{x}} + \epsilon$, where the design matrix $\mathbf{A} \in \mathbb{R}^{500\times 5000}$ and the noise term $\epsilon \in \mathbb{R}^n$ are randomly generated from normal distributions. The ground truth $\bar{\mathbf{x}} \in \mathbb{R}^n$ contains 10% of the signals, which are evenly partitioned into 5 groups. Specifically, within each group, we first continuously assign the same value for all the signals; and then, we randomly pick $\{0\%, 1\%, 2\%, 5\%, 10\%\}$ of the signals and change their signs to the opposite. The regularization parameters λ_1 and λ_2 are chosen from the interval $[10^{-4}\bar{\lambda}, 0.9\bar{\lambda}]$ using five-fold cross-validation for both the AFL and the fused Lasso. We then evaluate the models on a 100 i.i.d. samples testing set. The

SLEP package [7, 9] is utilized to solve the fused Lasso problem. We report the averaged prediction performance of 10 replications in Table 1.

We observe from Table 1 that the AFL approach provides better prediction performance than the fused Lasso in most cases. If the ground truth $\bar{\mathbf{x}}$ does not contain too many opposite adjacent signals, both AFL and the fused Lasso can recover the original signal accurately. However, when the number of opposite signals increases, AFL outperforms the fused Lasso significantly. The reason is that, with the AFL penalty, the model tends to select those highly similar adjacent features even if their signs are different. Therefore, the AFL approach is more robust than the fused Lasso in such cases.

Table 1: Averaged prediction performance of AFL and fused Lasso on synthetic data (standard deviation is shown in the bracket). FL refers to the fused Lasso. MSE refers to the mean squared error. Corr_X is the Pearson correlation between the model $\mathbf{x}$ and the ground truth $\bar{\mathbf{x}}$.

Neg%	Method	MSE_Y	MSE_X	Corr_X
0%	AFL	0.0001 (0.00)	0.0000 (0.00)	1.00 (0.00)
	FL	0.0003 (0.00)	0.0000 (0.00)	1.00 (0.00)
1%	AFL	0.0157 (0.02)	0.0000 (0.00)	1.00 (0.00)
	FL	0.0051 (0.00)	0.0000 (0.00)	1.00 (0.00)
2%	AFL	0.0179 (0.01)	0.0000 (0.00)	1.00 (0.00)
	FL	0.0227 (0.01)	0.0000 (0.00)	1.00 (0.00)
5%	AFL	15.16 (11.09)	0.0029 (0.00)	0.98 (0.01)
	FL	51.75 (23.55)	0.0103 (0.00)	0.92 (0.04)
10%	AFL	86.32 (28.21)	0.0200 (0.00)	0.81 (0.03)
	FL	125.98 (19.85)	0.0242 (0.00)	0.78 (0.01)

6.2 Real-world Applications

6.2.1 GLT1D1 Data Study

In this study, we evaluate the AFL approach on a real-world GWAS data called GLT1D1. The data set containing 210 samples and 1,782 SNPs [28]. The major objectives in this study are predicting the gene expression level of GLT1D1 as well as identifying disease risk SNPs. To construct our predictive models, we randomly pick 2/3 of the samples to form the training set and use the same method in § 6.1.2 to choose the best parameters. We compare the AFL model with the fused Lasso on the remaining 1/3 of the data. The averaged results of 10 replications are summarized in Table 2.

Table 2 shows that the AFL approach achieves better prediction performance in terms of MSE. In addition, our model selects a smaller number of SNPs, which demonstrates the need of considering the absolute values in GWAS due to the ambiguity choice of reference allele during genotype coding.

Table 2: Averaged prediction performance of AFL and fused Lasso on GLT1D1 Data. "# of nonzeros" refers to the number of nonzero regression coefficients.

Method	MSE	# of nonzeros
AFL	0.8952 (0.05)	15.25 (14.57)
FL	0.9182 (0.07)	34.78 (48.07)

6.2.2 ADNI WGS Data Study

In this study, we evaluate the AFL model on the Alzheimer's Disease Neuroimaging Initiative (ADNI) whole genome sequence (WGS) data. Particularly, we investigate imaging genetics associations between imaging phenotypes and SNPs (within the 19th chromosome) using the regression model with the AFL penalty. We follow the procedure in [24] to process the SNP data and the data set contains 717 subjects and 131,670 SNPs. The baseline entorhinal cortex (EC) volume and hippocampal (HIPP) volume are chosen to be the responses, as these are two major brain regions affected by the Alzheimer's disease (AD).

Comparison of Prediction Performance
We first compare the predictive performance of the AFL model with the fused Lasso. We randomly pick 90% of the samples to form the training set and the remaining 10% of the samples to form the testing set. We perform five-fold cross-validation to choose the best regularization parameters from the interval $[10^{-4}\bar{\lambda}, 0.9\bar{\lambda}]$. We report the mean squared error and the number of nonzero coefficients of 10 replications in Table 3.

Table 3 shows that both approaches achieve similar predictive performance in terms of MSE. Specifically, in EC task, AFL achieves a slightly lower MSE by selecting a smaller number of features. In HIPP task, AFL selects more features than the fused Lasso. We take a careful look at the SNPs identified by AFL. The AFL detects several SNPs located in three gene including PVRL2 (rs12972156, rs12972970, rs34342646), APOE (rs769449, rs769450, rs429358) and APOC1 (rs12721051, rs56131196, rs4420638) and assign them into correct groups. Note that the sign of the model coefficient of SNP rs769450 is different from its two adjacent SNPs (i.e., rs769449 and rs429358); the fused Lasso approach fails to group those three SNPs appropriately.

Table 3: Averaged prediction performance of AFL and fused Lasso on ADNI WGS Data.

	Method	MSE	# of nonzeros
EC	AFL	0.9422 (0.13)	19.4 (18.51)
	FL	0.9440 (0.12)	66.3 (23.78)
HIPP	AFL	0.9804 (0.13)	81.22 (86.21)
	FL	0.9996 (0.13)	58.22(23.26)

Detecting Risk Genetic Factors using AFL
Inspired by the idea of interaction testing introduced in [2], we finally conduct a study on detecting AD risk genetic factors with the AFL model. Specifically, on Chromosome 19, we first calculate the Pearson correlation between each coded SNP and the response phenotype vector. Then, we plug the correlation coefficients vector into our model (1). To identify the most association SNPs, we vary the regularization parameters and record each model.

Figure 3 shows the study results of EC and HIPP. In the experiment, we can observe that the AFL model can successfully capture AD risk genes including PVRL2 [10], TOMM40 [12, 6, 11], APOE [10, 12, 11, 19] and APOC1 [27, 19]. Moreover, the AFL is capable of performing automatic feature grouping even when the signs are different, e.g., rs769449, rs769450 and rs429358 in APOE exhibit high similarity in absolute values. However, the fused Lasso fails to correctly group SNPs like rs769450 since their signals are

Table 4: Statistical scores of selected SNPs on Chr.19. P-EC refers to the p-value associated with the EC task. P-HIPP refers to the p-value associated with the HIPP task. OR refers to the odds ratio associated with MCI&AD.

RS_ID	Gene	P-EC	P-HIPP	OR
rs12972156	PVRL2	1.03E-04	1.23E-05	1.947
rs12972970	PVRL2	1.24E-04	9.98E-06	1.984
rs34342646	PVRL2	1.18E-04	9.51E-06	1.809
rs283815	PVRL2	1.98E-04	1.17E-03	1.436
rs6857	PVRL2	8.07E-06	2.05E-06	1.914
rs76692773	PVRL2∼TOMM40	3.86E-01	2.64E-01	0.912
rs71352238	PVRL2∼TOMM40	9.20E-05	1.32E-05	1.767
rs184017	TOMM40	2.72E-05	8.31E-04	1.414
rs2075650	TOMM40	5.33E-04	3.15E-04	1.791
rs157581	TOMM40	5.43E-05	1.39E-03	1.436
rs34095326	TOMM40	4.14E-02	6.25E-02	1.511
rs34404554	TOMM40	1.59E-04	4.42E-05	1.842
rs11556505	TOMM40	1.60E-04	4.23E-05	1.857
rs157582	TOMM40	8.06E-05	1.96E-03	1.435
rs59007384	TOMM40	5.20E-05	5.13E-04	1.541
rs769449	APOE	1.54E-05	3.30E-06	2.646
rs769450	APOE	9.99E-03	2.87E-03	0.897
rs429358	APOE	2.13E-08	2.50E-07	2.409
rs10414043	APOE∼APOC1	1.49E-05	3.17E-05	2.447
rs7256200	APOE∼APOC1	1.96E-05	6.68E-05	2.447
rs483082	APOE∼APOC1	1.30E-04	1.55E-03	1.690
rs12721051	APOC2	1.73E-07	8.62E-06	1.914
rs56131196	APOC3	3.44E-08	5.11E-05	1.739
rs4420638	APOC4	3.40E-08	6.77E-05	1.712
rs78959900	APOC1	3.28E-02	1.27E-01	0.899
rs73052341	APOC1	4.65E-05	3.97E-05	1.978

different. In Table 4, we further present some statistical scores of SNPs selected by the AFL model, including the p-value[1] (P) and odds ratio (OR) association score. We can observe that most of the selected SNPs achieve high statistical significance.

7. CONCLUSIONS

In this paper, we study a regularized learning model based on absolute fused Lasso penalty. The AFL penalty encourages sparsity in the coefficients as well as penalizes differences of successive coefficients' magnitudes. Due to the non-convexity of the proposed model, we propose to use the DC programming to solve it. At each DC iteration, we solve a convex regularized sub-problem via the proximal algorithm. The proximal algorithm iteratively solves a proximal operator problem and adopts the Barzilai-Borwein rule for line search. One of our main technical contributions is to develop a highly efficient algorithm to solve the proximal operator problem via a Euclidean projection based on a novel restart technique. Experimental results on both synthetic and real-world data demonstrate the effectiveness and efficiency of the proposed algorithm.

8. ACKNOWLEDGMENTS

This work was supported in part by research grants from NIH (R01 LM010730 and RF1 AG051710) and NSF (IIS-0953662 and III-1421057).

References

[1] F. Bach, R. Jenatton, J. Mairal, G. Obozinski, et al. Structured sparsity through convex optimization. *Statistical Science*, 27(4):450–468, 2012.

[1]Those p-values are obtained from Pearson correlation analysis between SNPs and the selected imaging phenotype.

[2] J. Bien, N. Simon, R. Tibshirani, et al. Convex hierarchical testing of interactions. *The Annals of Applied Statistics*, 9(1):27–42, 2015.

[3] H. D. Bondell and B. J. Reich. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR. *Biometrics*, 64(1):115–123, 2008.

[4] J. Friedman, T. Hastie, H. Höfling, R. Tibshirani, et al. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

[5] 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.

[6] R. J. Guerreiro and J. Hardy. TOMM40 association with Alzheimer disease: tales of APOE and linkage disequilibrium. *Archives of neurology*, 69(10):1243–1244, 2012.

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

[8] J. Liu, K. Wang, S. Ma, and J. Huang. Regularized regression method for genome-wide association studies. *BMC Proceedings*, 5(9):1–5, 2011.

[9] J. Liu, L. Yuan, and J. Ye. An efficient algorithm for a class of fused lasso problems. In *Proceedings of the 16th ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining*, pages 323–332. ACM, 2010.

[10] M. W. Logue et al. A comprehensive genetic association study of Alzheimer disease in african americans. *Archives of neurology*, 68(12):1569–1579, 2011.

[11] D. M. Lyall et al. Alzheimer's disease susceptibility genes APOE and TOMM40, and brain white matter integrity in the lothian birth cohort 1936. *Neurobiology of aging*, 35(6):1513–e25, 2014.

[12] A. Maruszak et al. TOMM40 rs10524523 polymorphism's role in late-onset Alzheimer's disease and in longevity. *Journal of Alzheimer's Disease*, 28(2):309–322, 2012.

[13] D. E. Reich, M. Cargill, S. Bolk, J. Ireland, P. C. Sabeti, D. J. Richter, T. Lavery, R. Kouyoumjian, S. F. Farhadian, R. Ward, et al. Linkage disequilibrium in the human genome. *Nature*, 411(6834):199–204, 2001.

[14] P. D. Tao and L. T. H. An. Convex analysis approach to dc programming: Theory, algorithms and applications. *Acta Mathematica Vietnamica*, 22(1):289–355, 1997.

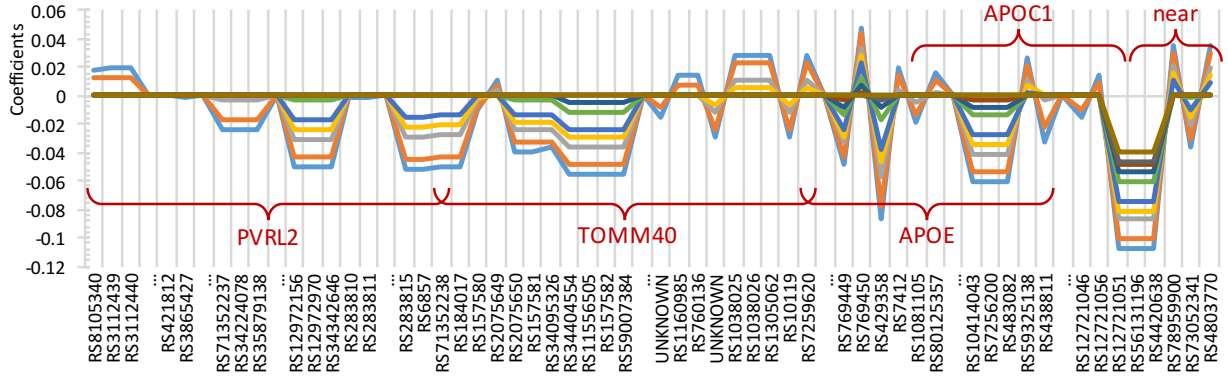
[15] P. D. Tao et al. Duality in dc (difference of convex functions) optimization. subgradient methods. In *Trends in Mathematical Optimization*, pages 277–293. Springer, 1988.

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

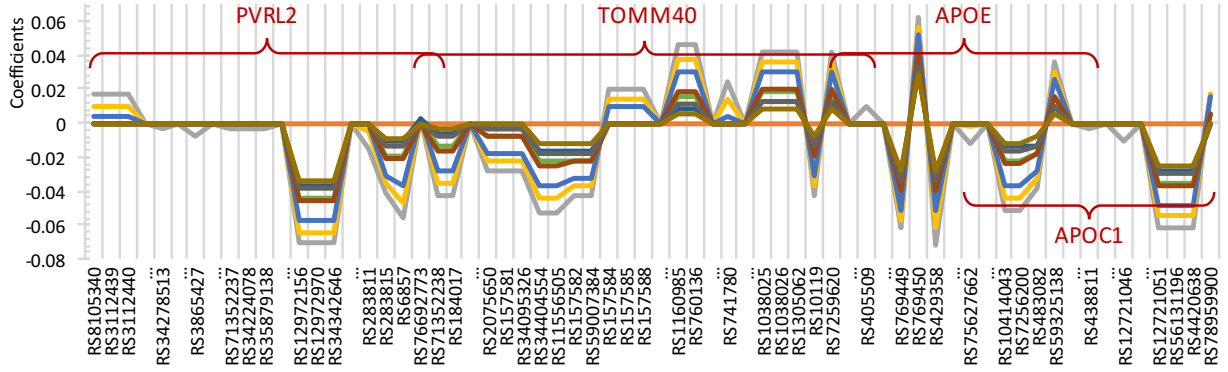
[17] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[18] R. Tibshirani and P. Wang. Spatial smoothing and hot spot detection for cgh data using the fused lasso. *Biostatistics*, 9(1):18–29, 2008.

[19] B. Tycko et al. APOE and APOC1 promoter polymorphisms and the risk of Alzheimer disease in african



(a) Entorhinal



(b) Hippocampus

Figure 3: Regression coefficients learned by each AFL model. Each color in the graph represents a learned model based on a pair of regularizers (λ_1, λ_2). SNPs (named by RS_IDs) are presented in their order on Chr.19. "..." indicates the gaps between SNPs. AD risk genes are marked in red.

american and caribbean hispanic individuals. *Archives of neurology*, 61(9):1434–1439, 2004.

[20] Y. Wang, S. Wang, J. Tang, H. Liu, and B. Li. PPP: Joint pointwise and pairwise image label prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

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

[22] S. Yang, Z. Lu, X. Shen, P. Wonka, and J. Ye. Fused multiple graphical lasso. *SIAM Journal on Optimization*, 25(2):916–943, 2015.

[23] S. Yang, L. Yuan, Y.-C. Lai, X. Shen, P. Wonka, and J. Ye. Feature grouping and selection over an undirected graph. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 922–930. ACM, 2012.

[24] T. Yang, J. Wang, Q. Sun, D. P. Hibar, N. Jahanshad, L. Liu, Y. Wang, L. Zhan, P. Thompson, and J. Ye. Detecting genetic risk factors for Alzheimer's disease in whole genome sequence data via Lasso screening. In *IEEE Intl. Symposium on Biomedical Imaging*, 2015.

[25] T. Yang, X. Zhao, B. Lin, T. Zeng, S. Ji, and J. Ye. Automated gene expression pattern annotation in the mouse brain. In *Pacific Symposium on Biocomputing*, page 144, 2015.

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

[27] Q. Zhou, F. Zhao, Z.-p. Lv, C.-g. Zheng, W.-d. Zheng, L. Sun, N.-n. Wang, S. Pang, F. M. de Andrade, M. Fu, et al. Association between APOC1 polymorphism and AlzheimerâĂŹs disease: A case-control study and meta-analysis. *PloS one*, 9(1):e87017, 2014.

[28] Y. Zhu, X. Shen, and W. Pan. Simultaneous grouping pursuit and feature selection over an undirected graph. *Journal of the American Statistical Association*, 108(502):713–725, 2013.

[29] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

APPENDIX

In this supplement, we present all of the details we mentioned in the main text.

A. SETTING FOR EXAMPLE 1

The synthetic data set is generated via $y = Ax + \epsilon$, where the data matrix $A \in \mathbb{R}^{500\times500}$ and the noise term ϵ are randomly generated from normal distributions. The ground truth x is designed as:

- $x_{1,\ldots,100,401,\ldots,500} = 0$,

- $x_{101,\ldots,190,201,\ldots,290,301,\ldots,390} = 1$,
- $x_{191,\ldots,200,291,\ldots,300,391,\ldots,400} = -1$.

B. DC PROGRAMMING FOR SOLVING AFL

The AFL formulation in Eq. (1) is a non-convex optimization problem. We propose to use the DC programming to solve it. By noting that

$$||x_i| - |x_{i+1}|| = |x_i + x_{i+1}| + |x_i - x_{i+1}| - (|x_i| + |x_{i+1}|),$$

we decompose the objective function in Eq. (1) into the difference of the following two functions:

$$f_1(\mathbf{x}) = \text{loss}(\mathbf{x}) + \lambda_1\|\mathbf{x}\|_1 + \lambda_2\sum_{i=1}^{p-1}(|x_i + x_{i+1}| + |x_i - x_{i+1}|),$$

$$f_2(\mathbf{x}) = \lambda_2\sum_{i=1}^{p-1}(|x_i| + |x_{i+1}|).$$

Denote the affine minorization of $f_2(\mathbf{x})$ as $f_2^k(\mathbf{x}) = f_2(\mathbf{x}^k) + \langle \mathbf{x} - \mathbf{x}^k, \partial f_2(\mathbf{x}^k)\rangle$, where $\langle\cdot,\cdot\rangle$ refers to the inner product. Then the DC programming solves problem (1) by iteratively solving:

$$\min_{\mathbf{x}\in\mathbb{R}^p} f_1(\mathbf{x}) - f_2^k(\mathbf{x}). \tag{24}$$

Since $\langle \mathbf{x}^k, \partial f_2(\mathbf{x}^k)\rangle$ is a constant, problem (24) is equivalent to:

$$\min_{\mathbf{x}\in\mathbb{R}^p} f_1(\mathbf{x}) - \langle \mathbf{x}, \partial f_2(\mathbf{x}^k)\rangle. \tag{25}$$

and let $\mathbf{c}^k = \partial f_2(\mathbf{x}^k)$, problem (24) can be rewritten as:

$$\min_{\mathbf{x}\in\mathbb{R}^p} \text{loss}(\mathbf{x}) - (\mathbf{c}^k)^T\mathbf{x} + \lambda_1\|\mathbf{x}\|_1 + \lambda_2\sum_{i=1}^{p-1}(|x_i + x_{i+1}| + |x_i - x_{i+1}|). \tag{26}$$

Note that

$$c_i^k = \lambda_2 d_i \,\text{sgn}\,(x_i^k), \tag{27}$$

where $d_1 = d_p = 1, d_i = 2, 2 \le i \le p-1$; $\text{sgn}(\cdot)$ is the signum function. In addition, since $\max(|x_i|, |x_{i+1}|) = \frac{1}{2}(|x_i + x_{i+1}| + |x_i - x_{i+1}|)$, problem (26) is equivalent to

$$\min_{\mathbf{x}\in\mathbb{R}^p} \text{loss}(\mathbf{x}) - (\mathbf{c}^k)^T\mathbf{x} + \lambda_1\|\mathbf{x}\|_1 + 2\lambda_2\sum_{i=1}^{p-1}\max(|x_i|, |x_{i+1}|).$$

C. PROOF FOR LEMMA 1

Proof: We prove these properties of the proximal operator problem (10) as follows.

i) If $u_i \ge 0$ and $x_i^* < 0$, we can construct $\tilde{\mathbf{x}}^*$ as follows:

$$\tilde{x}_i^* = 0, \tilde{x}_j^* = x_j^*, \forall j \ne i.$$

It can easily be shown that $\phi(\tilde{\mathbf{x}}^*) < \phi(\mathbf{x}^*)$. This contradicts with the fact that $\mathbf{x}^*$ is the minimizer to (10). If $u_i \ge 0$ and $x_i^* > u_i$, we can construct $\tilde{\mathbf{x}}^*$ as follows:

$$\tilde{x}_i^* = u_i, \tilde{x}_j^* = x_j^*, \forall j \ne i.$$

It can easily be shown that $\phi(\tilde{\mathbf{x}}^*) < \phi(\mathbf{x}^*)$. This contradicts with the fact that $\mathbf{x}^*$ is the minimizer to (10).

ii) This property can be proved in a similar way as i).

iii) Let $\tilde{\mathbf{x}}^* = \pi_\lambda(|\mathbf{u}|)$. We have

$$\begin{aligned}\phi(\text{sgn}(\mathbf{u})\odot\tilde{\mathbf{x}}^*) &= \frac{1}{2}\|\,\text{sgn}(\mathbf{u})\odot\tilde{\mathbf{x}}^* - \mathbf{u}\|^2 \\ &\quad + \lambda\sum_{i=1}^{p-1}\max(|\,\text{sgn}(u_i)\tilde{x}_i^*|, |\,\text{sgn}(u_{i+1})\tilde{x}_{i+1}^*|) \\ &= \frac{1}{2}\|\,\text{sgn}(\mathbf{u})\odot(\tilde{\mathbf{x}}^* - |\mathbf{u}|)\|^2 \\ &\quad + \lambda\sum_{i=1}^{p-1}\max(|\tilde{x}_i^*|, |\tilde{x}_{i+1}^*|) \\ &= \frac{1}{2}\|\tilde{\mathbf{x}}^* - |\mathbf{u}|\|^2 + \lambda\sum_{i=1}^{p-1}\max(|\tilde{x}_i^*|, |\tilde{x}_{i+1}^*|).\end{aligned}$$

Since $\tilde{\mathbf{x}}^* = \pi_\lambda(|\mathbf{u}|)$ and the minimizer is unique, it follows that $\text{sgn}(\mathbf{u})\odot\tilde{\mathbf{x}}^*$ needs to minimize $\phi(\mathbf{x})$.

iv) We only focus on the case $u_i \ge u_{i+1} \ge 0$ in the proof and the results can be generated to the rest the cases using property iii). With properties i) and ii), we have $u_i \ge x_i^* \ge 0$ and $u_{i+1} \ge x_{i+1}^* \ge 0$. If this property does not hold, we have:

$$u_i \ge u_{i+1} \ge x_{i+1}^* > x_i^* \ge 0. \tag{28}$$

Next, we show that $x_{i+1}^* > x_i^*$ leads to a contradiction. With a non-negative ϵ and assuming $x_{i+1}^* - \epsilon > x_i^*$, we construct $\bar{\mathbf{x}}^*$ and $\tilde{\mathbf{x}}^*$ as follows:

$$\bar{x}_{i+1}^* = x_{i+1}^* - \epsilon, \bar{x}_j^* = x_j^*, \forall j \ne i+1, \tag{29}$$

$$\tilde{x}_i^* = x_{i+1}^* + \epsilon, \tilde{x}_j^* = x_j^*, \forall j \ne i. \tag{30}$$

where the $i+1$ entry of $\mathbf{x}^*$ is decreased by ϵ in constructing $\bar{\mathbf{x}}^*$ and the i entry of $\mathbf{x}^*$ is increased by ϵ in constructing $\tilde{\mathbf{x}}^*$. Denote

$$d = -\frac{1}{2}(x_{i+1}^* - u_{i+1})^2 + \frac{1}{2}(x_{i+1}^* - \epsilon - u_{i+1})^2.$$

If $x_{i+1}^* < x_{i+2}^*$, we have

$$\phi(\bar{\mathbf{x}}) - \phi(\mathbf{x}^*) = d - \lambda\epsilon. \tag{31}$$

If $x_{i+1}^* - \epsilon > x_{i+2}^*$, we have

$$\phi(\bar{\mathbf{x}}) - \phi(\mathbf{x}^*) = d - 2\lambda\epsilon. \tag{32}$$

If $x_{i+1}^* \ge x_{i+2}^* \ge x_{i+1}^* - \epsilon$, we have

$$\phi(\bar{\mathbf{x}}) - \phi(\mathbf{x}^*) \le d - \lambda\epsilon. \tag{33}$$

$$\phi(\bar{\mathbf{x}}) - \phi(\mathbf{x}^*) \ge d - 2\lambda\epsilon. \tag{34}$$

In summary, we have

$$\begin{aligned}\phi(\bar{\mathbf{x}}) - \phi(\mathbf{x}^*) \le & g_1(\epsilon) = -\frac{1}{2}(x_{i+1}^* - u_{i+1})^2 \\ &+ \frac{1}{2}(x_{i+1}^* - \epsilon - u_{i+1})^2 - \lambda\epsilon.\end{aligned} \tag{35}$$

Similarly, we have

$$\begin{aligned}\phi(\tilde{\mathbf{x}}) - \phi(\mathbf{x}^*) \le g_2(\epsilon) = & -\frac{1}{2}(x_i^* - u_i)^2 \\ &+ \frac{1}{2}(x_i^* + \epsilon - u_i)^2 + \lambda\epsilon.\end{aligned} \tag{36}$$

It is hard to directly prove either $g_1(\epsilon)$ or $g_2(\epsilon)$ is negative in the case of (28). To arrive at the contradiction,

we let

$$\begin{aligned}\mathcal{G}(\epsilon) &= g_1(\epsilon) + g_2(\epsilon)\\ &= -\frac{1}{2}(x^*_{i+1} - u_{i+1})^2 + \frac{1}{2}(x^*_{i+1} - \epsilon - u_{i+1})^2\\ &\quad - \frac{1}{2}(x^*_i - u_i)^2 + \frac{1}{2}(x^*_i + \epsilon - u_i)^2\end{aligned} \tag{37}$$

The derivative of $\mathcal{G}(\epsilon)$ is

$$\mathcal{G}'(\epsilon) = 2\epsilon + (u_{i+1} - u_i) + (x^*_i - x^*_{i+1}). \tag{38}$$

Making use of (28), we can arrive at $\mathcal{G}'(\epsilon) < 0$ when

$$\epsilon \in (0, \frac{x^*_{i+1} - x^*_i}{2}). \tag{39}$$

For any ϵ satisfying (39), we have $\mathcal{G}(\epsilon) < 0$, since $\mathcal{G}(0) = 0$ and $\mathcal{G}'(\epsilon) < 0$. Therefore, there exists ϵ that satisfies (39). Hence

$$(\phi(\bar{\mathbf{x}}^*) - \phi(\mathbf{x}^*)) + (\phi(\tilde{\mathbf{x}}^*) - \phi(\mathbf{x}^*)) < 0. \tag{40}$$

This leads to the fact that at least one of the following two inequalities holds:

$$(\phi(\bar{\mathbf{x}}^*) - \phi(\mathbf{x}^*)) < 0,$$

$$(\phi(\tilde{\mathbf{x}}^*) - \phi(\mathbf{x}^*)) < 0.$$

This contradicts with the fact that $\mathbf{x}^*$ is the minimizer to (10). Therefore, we cannot have $x^*_{i+1} > x^*_i$ in the case $u_i \geq u_{i+1} \geq 0$.

v) This property can be proved in a similar way as iv).

This ends the proof to Lemma 1. □

Based on Lemma 1, we also have the following remark that summarizes the properties of w:

Remark. *$w_i = 2$ indicates $1 < i < p, u_{i-1} < u_i \leq u_{i+1}$.*
$w_i = 1$ holds in one of the following four cases:
1) $i = 1, u_1 \geq u_2$;
2) $i = p, u_{p-1} < u_p$;
3) $1 < i < p, u_i \geq u_{i+1}, u_i \leq u_{i-1}$;
4) $1 < i < p, u_i < u_{i+1}, u_i > u_{i-1}$.
$w_i = 0$ holds in one of the following three cases:
1) $i = 1, u_1 < u_2$;
2) $i = p, u_{p-1} \geq u_p$;
3) $1 < i < p, u_i < u_{i+1}, u_i \geq u_{i-1}$.
In addition, it is easy to get that $\sum_{i=1}^{p} w_i = p - 1$.

D. PROOF FOR THEOREM 2

Proof: According to Lemma 1 i) and ii), we have that the optimal solution to (10) is non-negative, i.e., $\mathbf{x}^* \geq 0$. Incorporating the definition of R in (11) and Lemma 1 iv) and v), we have $R\mathbf{x}^* \leq 0$. Therefore, we have $\mathbf{x}^* \in P$, where P is defined in (14). It is easy to verify that P is a closed convex and nonempty polyhedron. Thus $\mathbf{x}^* = \pi_\lambda(\mathbf{u})$ is the optimal solution to

$$\min_{\mathbf{x}\in P} \left\{ \frac{1}{2}\|\mathbf{x} - \mathbf{u}\|^2 + \lambda \sum_{i=1}^{p-1} \max(|x_i|, |x_{i+1}|) \right\}.$$

Making use of the definitions of R and $\mathbf{w}$ in (11) and (12), $\forall \mathbf{x} \in P$, we have

$$\sum_{i=1}^{p-1} \max(|x_i|, |x_{i+1}|) = \sum_{i=1}^{p} w_i x_i.$$

Therefore, $\mathbf{x}^* = \pi_\lambda(\mathbf{u})$ is the optimal solution to the problem:

$$\min_{\mathbf{x}\in P} \left\{ \frac{1}{2}\|\mathbf{x} - \mathbf{u}\|^2 + \lambda \sum_{i=1}^{p} w_i x_i \right\}. \tag{41}$$

Incorporating (13), we can easily verify that, $\pi^P_\lambda(\mathbf{v})$, the optimal solution to(15) is also the optimal solution to (41). Thus, (16) holds. □

E. PROOF FOR THEOREM 3

Proof: We prove (19) by the technique of KKT optimality conditions.

By introducing the dual variables $\mathbf{w} \in \mathbb{R}^p$ for the inequality $\mathbf{x} \geq 0$, and $\mathbf{z} \in \mathbb{R}^{p-1}$ for the inequality $R\mathbf{x} \leq 0$, we can write the Lagrangian of (15) as:

$$\mathcal{L}(\mathbf{x}, \mathbf{w}, \mathbf{z}) = \frac{1}{2}\|\mathbf{x} - \mathbf{v}\|^2 - \mathbf{w}^T\mathbf{x} + \mathbf{z}^T R\mathbf{x}. \tag{42}$$

The inequality constraint functions in (15) are affine, and thus Slater's condition holds, which indicates strong duality. Let $\bar{\mathbf{x}}$ and $(\bar{\mathbf{w}}, \bar{\mathbf{z}})$ be any primal and dual optimal points with zero gap for (15). The KKT optimality conditions require the following necessary and sufficient conditions:

$$\begin{aligned}\bar{\mathbf{x}} &\geq 0, &(43)\\ R\bar{\mathbf{x}} &\leq 0, &(44)\\ \bar{\mathbf{w}} &\geq 0, &(45)\\ \bar{\mathbf{z}} &\geq 0, &(46)\\ \bar{\mathbf{x}} &= \mathbf{v} + \bar{\mathbf{w}} - R\bar{\mathbf{z}}. &(47)\end{aligned}$$

Following a similar analysis, we introduce the dual variable $\mathbf{z} \in \mathbb{R}^{p-1}$ for the inequality $R\mathbf{x} \leq 0$, and write the Lagrangian of (18) as:

$$\mathcal{L}(\mathbf{x}, \mathbf{z}) = \frac{1}{2}\|\mathbf{x} - \mathbf{v}\|^2 + \mathbf{z}^T R\mathbf{x}. \tag{48}$$

Let $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{z}}$ be any primal and dual optimal points with zero gap for (18). The KKT optimality conditions requires the following necessary and sufficient conditions:

$$\begin{aligned}R\tilde{\mathbf{x}} &\leq 0, &(49)\\ \tilde{\mathbf{z}} &\geq 0, &(50)\\ \tilde{\mathbf{x}} &= \mathbf{v} - R\tilde{\mathbf{z}}. &(51)\end{aligned}$$

Let

$$\begin{aligned}\mathbf{x}^* &= \max(\tilde{\mathbf{x}}, 0), &(52)\\ \mathbf{w}^* &= \max(\tilde{\mathbf{x}}, 0) - \tilde{\mathbf{x}}, &(53)\\ \mathbf{z}^* &= \tilde{\mathbf{z}}. &(54)\end{aligned}$$

Next, we show that $\mathbf{x}^*$ and $(\mathbf{w}^*, \mathbf{z}^*)$ satisfy the KKT conditions (44)-(47). It is easy to verify the relationships in formulations (44), (46)-(47). $R\mathbf{x}^* \leq 0$ holds as: 1) $R\tilde{\mathbf{x}} \leq 0$, 2) $\mathbf{x}^* = \max(\tilde{\mathbf{x}}, 0)$, and 3) each row of R only contains two entries 1 and -1. As the objectives of (15) and (18) are strictly convex, $\bar{\mathbf{x}}$ and $\tilde{\mathbf{x}}$ are both unique. Therefore, it follows from (53) that (19) holds. □