Identifying Genetic Risk Factors for Alzheimer's Disease via Shared Tree-guided Feature Learning across Multiple Tasks

Weizhong Zhang, Tingjin Luo, Shuang Qiu, Jieping Ye, *Senior Member, IEEE*
Deng Cai, *Member, IEEE,* Xiaofei He, *Senior Member, IEEE,* and Jie Wang

Abstract—The genome-wide association study (GWAS) is a popular approach to identify disease-associated genetic factors for Alzhemer's Disease (AD). However, it remains challenging because of the small number of samples, very high feature dimensionality and complex structures. To accurately identify genetic risk factors for AD, we propose a novel method based on an in-depth exploration of the hierarchical structure among the features and the commonality across related tasks. Specifically, we first extract and encode the tree hierarchy among features; then, we integrate the tree structures with multi-task feature learning (MTFL) to learn the shared features—that are predictive of AD—among related tasks simultaneously. Thus we can unify the strength of both the prior structure information and MTFL to boost the prediction performance. However, due to the highly complex regularizer that encodes the tree structure and the extremely high feature dimensionality, the learning process can be computationally prohibitive. To address this, we further develop a novel safe screening rule to quickly identify and remove the irrelevant features before training. Experiment results demonstrate that the proposed approach significantly outperforms the state-of-the-art in detecting genetic risk factors of AD and the speedup gained by the proposed screening can be several orders of magnitude.

Index Terms—Tree-structured group Lasso, Multi-task learning, Alzheimer's disease, Genome-wide association studies, Screening.

1 Introduction

The genome-wide association study (GWAS) [1], [2], [3] is an emerging technique to identify genetic risk factors by uncovering the associations between single nucleotide polymorphisms (SNPs) and quantitative traits, such as brain images of patients with Alzheimer's-disease (AD) [4]. In the past few years, GWAS has achieved great success in identifying genetic risk factors for many human diseases including diabetes [5], [6], heart abnormalities [7], [8] and Parkinson disease [9], [10], [11]. In this paper, we focus on the detection of genetic risk factors for AD.

Many early studies on GWAS are based on univariate analysis, which does not take into account the joint effect of multiple SNPs. Different from these univariate approaches, Lasso [12] is a popular multivariate technique [13], [14] that is capable of addressing the potential dependency among features. Due to its ℓ_1 sparsity-inducing regularizer, Lasso provides a set of candidate SNPs for the genetic risk factors. However, as the ℓ_1-regularization treats all SNPs equally, traditional Lasso based approaches do not take the feature structures into consideration. Recently, Yang et al. [15] proposed a non-convex model named Absolute Fused Lasso (AFL) model with a penalty, which encourages sparsity in the coefficients as well as their successive differences of absolute values, to identify the genetic risk factors related to AD. Nevertheless, AFL also cannot take feature structure into consideration.

Indeed, in many real-world applications, the features exhibit certain intrinsic structures, such as spatial or temporal smoothness, groups, and trees. Many recent studies [2], [3], [16] have shown that incorporating the *a prior* structure information can significantly improve the prediction performance and facilitate the identification of important features. Motivated by this observation, tree-structured group Lasso (TGL) [16], [17] has been employed to analyze the GWAS data by incorporating the tree hierarchy—that can be obtained by linkage disequilibrium (LD [18])—among SNPs, thereby providing superior performance in identifying AD-related SNPs.

Motivations: However, most existing work focuses on one task, e.g., one brain region of AD patients, while we may have the magnetic resonance imaging (MRI) data for multiple brain regions that can be exploited. Another line of research proposes to learn a set of shared SNPs across several related tasks under the framework of multi-task feature learning (MTFL) [19], [20], though they ignore the hierarchical tree structure among SNPs. To enhance the advantages of the *prior* tree structure and the potential commonality across related tasks in identifying AD-related

- *W. Zhang, D. Cai and X. He are with the State Key Lab of CAD&CG, College of Computer Science, Zhejiang University, 388 Yuhang Tang Road, Hangzhou, Zhejiang 310058, China. W. Zhang is also with Tencent AI Lab, Shenzhen, China.*
E-mail: zhangweizhongzju@gmail.com, dengcai@cad.zju.edu.cn, xiaofeihe@cad.zju.edu.cn
- *T. Luo, S. Qiu and J. Ye are with the Department of Computational Medicine and Bioinformatics, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA. T. Luo is also with the College of Science, National University of Defense Technology, Changsha, Hunan, China.*
E-mail: {tingjinluo,shuangqiu.umich,jieping}@gmail.com
- *Jie Wang is with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, 96 JinZhai Road, Baohe District, Hefei, Anhui, 230026, China.*
E-mail: jiewangustc@gmail.com

The first two authors contribute equally.

Manuscript received July 27, 2017; revised November 06, 2017.

SNPs, in this paper, we propose a novel—that is also the first—approach to learn a **S**hared **T**ree-structured sparse pattern across **M**ultiple related tasks (STM). We note that, although the work in [21] also proposes to combine TGL with MTFL, the tree structure is assumed on the multivariate output space, which is the major difference from the proposed STM.

Although the integration of the *prior* tree structure and MTFL may dramatically enhance the prediction performance of STM in identifying the AD-related SNPs, the optimization of STM may be computationally prohibitive due to the nonsmooth and highly complex regularizer. Moreover, the high feature dimensionality makes it even more challenging in applying STM to analyze GWAS data. To address this, we propose an efficient **H**ierachical **F**eature **S**creening rule (HFS) for STM to quickly identify the irrelevant features, which can be removed from the training phase, without losing any useful information. This leads to a significant speedup—that can be orders of magnitude—of the downstream optimization of STM. Moreover, HFS is *safe* as the models learned on the reduced data are identical to the ones learned on the full data.

The main contributions of this paper are thus:

- We propose a novel method named STM for accurately identifying genetic risk factors for AD, which can explore in-depth the hierarchical structure among the features and the commonality across related tasks.
- Since the learning process is very time consuming, we develop an effective screening rule called HFS for our proposed method, which can improve the time-efficiency by several orders of magnitude.
- Experiment results and the medical evaluation demonstrate that the proposed method is efficient and can greatly accelerate the discovery of causal variants for complex diseases.

The rest of this paper is organized as follows. In Section 2, we briefly review some basics of TGL, MTFL, and feature screening [17], [22], [23]. We then propose our method STM in Section 3. In Section 4, we derive the HFS screening rule in detail. We evaluate the proposed STM combined with HFS and other state-of-the-art methods on both synthetic and real-world data in terms of efficiency and prediction accuracy in Section 5. We give some concluding remarks in Section 6.

2 Preliminary

2.1 Tree-structured Group Lasso

TGL is a popular method that encodes the hierarchical structure—that can be represented by an *index tree*—among features. For a positive integer p, let $[p] = \{1, ..., p\}$. If $G_1, G_2 \subseteq [p]$, the inclusion $G_1 \subset G_2$ implies that G_1 is a proper subset of G_2. Then, an index tree is given as follows.

Definition 1. (Index Tree) *[24] For an index tree T of depth d, we denote the nodes of depth i by $T_i = \{G_1^i, ..., G_{n_i}^i\}$, where $n_0 = 1, G_1^0 = [p], G_j^i \subset [p]$, and $n_i \geq 1, \forall i \leq d$. In addition, the following conditions should be satisfied:*
(i) : $G_{j_1}^i \cap G_{j_2}^i = \emptyset, \forall i \leq d$ and $j_1 \neq j_2$ (different nodes in the same layer do not overlap).
(ii) : If G_j^i is a parent node of G_l^{i+1}, then $G_l^{i+1} \subset G_j^i$.

Figure 1 shows a simple index tree.

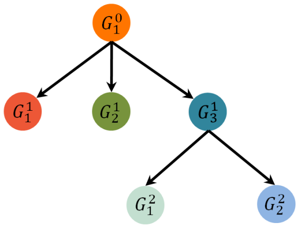


Fig. 1. Index tree example with $G_1^0 = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$, $G_1^1 = \{1, 2, 3\}, G_2^1 = \{4, 5\}, G_3^1 = \{6, 7, 8, 9\}$, $G_1^2 = \{6\}$, and $G_2^2 = \{7, 8, 9\}$.

Let $\|\cdot\|$ be the ℓ_2 norm of a given vector. If $G \subseteq [p]$, we denote the complement set of G with respect to $[p]$ by $\bar{G} = [p] \setminus G$. For a vector $\beta \in \mathbb{R}^p$, we denote the i-th component of β by $[\beta]_i$. Then, for $G \subseteq [p]$, we define β_G by $[\beta_G]_i = [\beta]_i$ if $i \in G$ and $[\beta_G]_i = 0$ otherwise. Suppose that the tree structure is available, then TGL takes the form of

$$\min_{\beta} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|^2 + \lambda \sum_{i=0}^{d} \sum_{j=1}^{n_i} w_j^i \|\beta_{G_j^i}\|, \tag{1}$$

where $\mathbf{y} \in \mathbb{R}^N$ is the response vector, $\mathbf{X} \in \mathbb{R}^{N \times p}$ is the data matrix, $\beta_{G_j^i}$ and w_j^i are the coefficients vector and positive weight (that is given) corresponding to node G_j^i, respectively, and $\lambda > 0$ being a regularization parameter.

We can see that, if $\beta_{G_j^i} = 0$ and G_k^l is a child node of G_j^i, that is, $G_k^l \subset G_j^i$, then $\beta_{G_k^l} = 0$.

2.2 Multi-task Feature Learning

MTFL is a powerful feature learning method that can exploit the commonality across related tasks. Given the training data $\{(\mathbf{X}_t, \mathbf{y}_t) : t = 1, ..., T\}$ for T tasks, where $\mathbf{X}_t \in R^{N_t \times p}$ is the data matrix of the t-th task consisting of N_t training samples and $\mathbf{y}_t \in \mathbb{R}^{N_t}$ is the corresponding response vector. Let $N = \sum_{t=1}^{T} N_t$ be the total number of samples, a popular MTFL model [25] takes the form of

$$\min_{\mathbf{B} \in \mathbb{R}^{N \times p}} \sum_{t=1}^{T} \frac{1}{2}||\mathbf{y}_t - \mathbf{X}_t\beta_t||^2 + \lambda||\mathbf{B}||_{2,1}, \tag{2}$$

where $\mathbf{B} = (\beta_1, ..., \beta_T)$ with β_t being the coefficient vector of the t-th task, $\|\mathbf{B}\|_{2,1} = \sum_{i=1}^{p} \|\beta^i\|$ with β^i being the i-th row of $\mathbf{B}$ and $\lambda > 0$ being a regularization parameter. Therefore, the regularizer encourages sparsity on the rows of $\mathbf{B}$ and then constrains all tasks to select a common set of features.

2.3 Feature Reduction by Screening

Screening refers to a suite of emerging techniques that can dramatically scale up sparse learning methods for big and complex data that were previously impossible. Essentially, screening aims to quickly identify the irrelevant features—that can be removed from the training phase—before the optimization algorithms are applied. The reduction of features can lead to substantial savings in computational time and memory usage by orders of magnitude.

Existing screening methods can be roughly divided into two categories: heuristic methods [26] and safe methods [17], [22], [27]. Briefly speaking, heuristic methods may mistakenly discard features that are supposed to be selected by the model. Therefore, heuristic methods usually have a post-processing procedure—by checking the Karush-Kuhn-Tucker (KKT) conditions [28]—to ensure that all relevant features are included in the candidate set. In contrast, for safe screening methods, the models learned on the reduced data are identical to the ones learned on the full data. Therefore, safe screening can significantly improve the efficiency of sparse learning methods without sacrificing accuracy.

In recent years, many efforts are devoted to developing efficient screening methods for a large set of sparse learning models, like Lasso, group Lasso, ℓ_1 regularized logistic regression, etc. To the best of our knowledge, existing screening rules are not applicable to our technically challenging STM formulation, which will be analyzed in detail in section 4. Thus, one of our major technical contributions is the proposed HFS screening rule for STM.

3 Our STM Formulation

In this section, we introduce the proposed STM (**S**hared **T**ree-guided feature learning across **M**ultiple tasks) model.

GWAS data sets usually do not have enough training samples as collecting genotype data is scarce, and the feature dimension can be much higher than the number of samples. Thus, the analysis of GWAS data remains challenging and many traditional techniques are inadequate. Recent studies [2], [3], [16]have shown that incorporating the hierarchical tree structure among features by the LD block information can significantly improve the prediction performance in identifying the genetic risk factors for AD. Moreover, we note that, multiple AD-related quantitative traits, such as the MRI data for multiple brain regions, are available. Enlightened by the aforementioned observations, we propose to detect the genetic risk factors for AD by learning multiple related tasks with a shared tree structure.

Suppose that we have T related tasks to learn. For the t-th task, let $\mathbf{X}_t \in \mathbb{R}^{N_t \times p}$ be the data matrix consisting of N_t samples, each of which is represented by a vector in $\mathbb{R}^p$, and let $\mathbf{y}_t \in \mathbb{R}^{N_t}$ be the response vector. Thus, we have $N = \sum_{t=1}^{T} N_t$ samples in total. To simplify notations, for a given matrix $\mathbf{M}$, let $\mathbf{m}_j$ and $\mathbf{m}^i$ be its j-th column and i-th row, respectively, and $\|\mathbf{M}\|_F$ be its Frobenius norm. If $G \subset [p]$, let $\mathbf{M}_G$ be the sub-matrix of $\mathbf{M}$ that consists of $\mathbf{m}^i$ with $i \in G$.

Supposing that the tree structure is available, the proposed STM takes the form of

$$\min_{\mathbf{B}\in\mathbb{R}^{p\times T}} \sum_{t=1}^{T} \frac{1}{2}||\mathbf{y}_t - \mathbf{X}_t\beta_t||^2 + \lambda \sum_{i=0}^{d}\sum_{j=1}^{n_i} w_j^i ||\mathbf{B}_{G_j^i}||_F, \quad (3)$$

where $\mathbf{B} = (\beta_1, \ldots, \beta_T)$, λ and w_j^i for $i = 0, \ldots, d$, $j = 1, \ldots, n_i$ are positive parameters.

Figure 2 illustrates how we impose a tree structure on the coefficient matrix $\mathbf{B}$. We can see that, each column β_t of $\mathbf{B}$ corresponds to the coefficient vector of the t-th task; each row β^l corresponds to the coefficients of the l-th features across all tasks. Thus, each node $G_j^i \subseteq [p]$ corresponds to a sub-matrix $\mathbf{B}_{G_j^i}$ of the coefficient matrix $\mathbf{B}$. This is the major difference that distinguishes STM from TGL, leading to significant technical challenges to the optimization. Therefore, the regularizer in STM [the second term in (3)] can encourage a shared tree structure among features across multiple tasks.

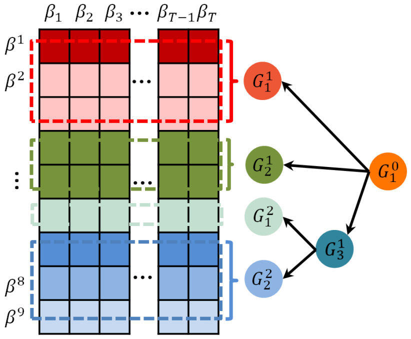


Fig. 2. An index tree shared by multiple tasks. The tree is the same as the one in Figure 1.

We note that the tree structure in [21] is assumed on the multivariate output space, which is the major difference compared to our STM.

4 The Proposed Screening Rule

In the last section, we present our proposed STM model for GWAS. However, due to the nonsmooth and highly complex regularizer and the extremely high dimensionality of the features, directly solving the optimization problem (3) is very time consuming. To this end, we develop an efficient screening rule called Hierarchically Feature Screening (HFS) for our model to improve the training efficiency.

Before presenting our method, we would like to point out that our HFS looks similar but differs in nature with the recent screening works [17]. Although we can rewrite our formulation (3) into the form of the problem in [17] by arranging the data matrices diagonally to form a big matrix, whose size would be as large as $\sum N_t \times pT(T-1$ times larger than the total size of all the data matrices). This leads to a sheer waste of memory and computational cost and makes [17] incapable in high dimensional and data intensive cases. Thus we cannot apply [17] to STM directly.

Similar to the works [17] and [22], our HFS is inspired by the KKT condition. It is mainly comprised of two steps: estimate the dual solution based on the solutions with a larger λ, then identify the inactive features based on the KKT condition. In this section, we will first derive the dual problem and the KKT condition of our model. Then, to make our key ideas in developing the rules more accessible, we develop the preliminary screening method when the optimal solution θ^* of the dual problem (4) is known. Finally, we design our screening rules for STM when the optimal solution θ^* of the dual problem is not available.

4.1 The Dual Problem and KKT Condition

To simplify notations, we denote $\mathbb{R}^{p\times T}$ by $\mathcal{H}$. For any $U_1, U_2 \in \mathcal{H}$, let $\langle U_1, U_2\rangle = \mathrm{tr}(U_1^T U_2)$. We first introduce the following useful result.

Lemma 1. *For any convex function $\psi(\mathbf{B}) : \mathcal{H} \to \mathbb{R}$, with $\psi(0) = 0$ and a fixed point $\zeta \in \mathcal{H}$. If we define $h(\mathbf{B}) = \psi(\mathbf{B}) - \langle \zeta, \mathbf{B} \rangle$, then the followings hold:*
(i) $\min_{\mathbf{B}\in\mathcal{H}} \mathrm{h}(\mathbf{B}) \geq 0 \Leftrightarrow \zeta \in \partial\psi(0)$,
(ii) $\min_{\mathbf{B}\in\mathcal{H}} \mathrm{h}(\mathbf{B}) = -\infty \Leftrightarrow \zeta \notin \partial\psi(0)$.

Proof. i) ($\Leftarrow$) Suppose that $\zeta \in \partial\psi(0)$. Then, we have

$$\psi(\mathbf{B}) - \psi(0) \geq \langle \zeta, \mathbf{B} \rangle, \Rightarrow h(\mathbf{B}) \geq \psi(0), \forall \mathbf{B} \in \mathcal{H}.$$

It implies that $\min_{\mathbf{B}\in\mathcal{H}} h(\mathbf{B}) \geq \psi(0) = 0$.

($\Rightarrow$) Suppose that $\min_{\mathbf{B}\in\mathcal{H}} h(\mathbf{B}) \geq 0$, we will show that $\zeta \in \partial\psi(0)$ by contradiction.

Suppose that $\zeta \notin \partial\psi(0)$. Then by the definition of subgradient, we can find $\mathbf{B}_0 \in \mathcal{H}$ such that

$$\psi(\mathbf{B}_0) - \psi(0) < \langle \zeta, \mathbf{B}_0 \rangle \Rightarrow h(\mathbf{B}_0) < \psi(0) = 0,$$

which contradicts the assumption that $\min_{\mathbf{B}\in\mathcal{H}} h(\mathbf{B}) \geq 0$. It implies that $\zeta \in \partial\psi(0)$. Thus the proof of (i) is completed.
ii) By part (i), we already have

$$\min_{\mathbf{B}\in\mathcal{H}} h(\mathbf{B}) < 0 \Leftrightarrow \zeta \notin \partial\psi(0).$$

So we only need to show that

$$\zeta \notin \partial\psi(0) \Rightarrow \min_{\mathbf{B}\in\mathcal{H}} h(\mathbf{B}) = -\infty.$$

Indeed, by part (i), $\zeta \notin \partial\psi(0)$ leads to

$$\exists \mathbf{B}_0 \in \mathcal{H}, \text{ such that } h(\mathbf{B}_0) < 0.$$

By noting that $h(t\mathbf{B}) = th(\mathbf{B})$ holds for any $t \geq 0$, we have

$$\lim_{t\to\infty} h(t\mathbf{B}_0) \to -\infty \Rightarrow \min_{\mathbf{B}} h(\mathbf{B}) = -\infty.$$

The proof is complete. □

Now with this lemma, we turn to derive the Lagrangian dual problem and the KKT condition for our model STM, which are given in Theorem 1 below.

Theorem 1. *For our model, the following holds:*
(i) *The Lagrangian dual problem can be written as*

$$\sup_{\theta}\{\frac{1}{2}||\boldsymbol{y}||^2 - \frac{1}{2}||\frac{\boldsymbol{y}}{\lambda} - \theta||^2 : \theta \in \mathcal{F}\}, \quad (4)$$

where the feasible region $\mathcal{F} = \{\theta = (\theta_1; ...; \theta_T) \in \mathbb{R}^N : M(\theta) = (X_1^T\theta_1, ..., X_T^T\theta_T) \in \partial\phi(0), \theta_t \in \mathbb{R}^{N_t}\}$.
(ii) *The KKT conditions are:*

$$\boldsymbol{y}_t - \mathbf{X}_t\beta_t^* = \lambda\theta_t^*, t = 1, ..., T, \quad (5)$$
$$M(\theta^*(\lambda)) = (X_1^T\theta_1^*, ..., X_T^T\theta_T^*) \in \partial\phi(\mathbf{B}^*), \quad (6)$$

where $\mathbf{B}^*$ and θ^* are the optimal solutions of the primal problem (3) and the dual problem (4) respectively.

Proof. We first introduce a sequence of new variables

$$\mathbf{z}_t = \mathbf{y}_t - \mathbf{X}_t\beta_t, t = 1, ..., T.$$

The primal problem (3) can then be rewritten as the constrained optimization problem below:

$$\min_{\mathbf{B}\in\mathcal{H}} \Big\{ \sum_{t=1}^{T} \frac{1}{2}||\mathbf{z}_t||^2 + \lambda\phi(\mathbf{B}) : \mathbf{z}_t = \mathbf{y}_t - \mathbf{X}_t\beta_t, t = 1, ..., T \Big\}.$$

For notational convenience, we let

$$f_1(\mathbf{B}) = \lambda(\phi(\mathbf{B}) - \sum_{t=1}^{T}\langle X_t^T\theta_t, \beta_t \rangle) = \lambda(\phi(\mathbf{B}) - \langle M(\theta), \mathbf{B} \rangle),$$

$$f_2(\mathbf{z}) = \sum_{t=1}^{T}(\frac{1}{2}||\mathbf{z}_t||^2 - \lambda\langle \theta_t, \mathbf{z}_t \rangle),$$

where $M(\theta) = (X_1^T\theta_1, ..., X_T^T\theta_T), \mathbf{z} = (\mathbf{z}_1; ...; \mathbf{z}_T)$, then the Lagrangian becomes

$$\begin{aligned} L(\mathbf{B}, \mathbf{z}, \theta) &= \sum_{t=1}^{T}\frac{1}{2}||\mathbf{z}_t||^2 + \lambda\phi(\mathbf{B}) + \lambda\sum_{t=1}^{T}\langle \theta_t, \mathbf{y}_t - \mathbf{X}_t\beta_t - \mathbf{z}_t \rangle \\ &= f_1(\mathbf{B}) + f_2(\mathbf{z}) + \lambda\sum_{t=1}^{T}\langle \theta_t, \mathbf{y}_t \rangle. \quad (7) \end{aligned}$$

From Lemma 1 and the fact that $f_1(0) = 0$, we have

$$\min_{\mathbf{B}\in\mathcal{H}} f_1(\mathbf{B}) = \begin{cases} 0, & \text{if } M(\theta) \in \partial\phi(0), \\ -\infty, & \text{otherwise.} \end{cases} \quad (8)$$

If $\mathbf{B}^* \in \arg\min_{\mathbf{B}} f_1(\mathbf{B})$, we have

$$0 \in \partial f_1(\mathbf{B}^*) \Leftrightarrow M(\theta) \in \partial\phi(\mathbf{B}^*), \quad (9)$$

which is one of the KKT conditions. In addition, we have

$$0 = \nabla_{\mathbf{z}^*} f_2(\mathbf{z}^*) \Rightarrow \mathbf{z}^* = \lambda\theta \text{ and } \min_{\mathbf{z}} f_2(\mathbf{z}) = -\frac{\lambda^2}{2}||\theta||^2. \quad (10)$$

Thus we get another KKT condition

$$\mathbf{y}_t - \mathbf{X}_t\beta_t^* = \lambda\theta_t^*, t = 1, ..., T. \quad (11)$$

By plugging (10) and (8) into (7), we can obtain the Lagrangian dual problem below

$$\sup_{\theta}\{\frac{1}{2}||\mathbf{y}||^2 - \frac{1}{2}||\frac{\mathbf{y}}{\lambda} - \theta||^2 : \theta \in \mathcal{F}\},$$
where $\mathcal{F} = \{\theta : M(\theta) = (X_1^T\theta_1, ..., X_T^T\theta_T) \in \partial\phi(0)\}$.

From formulations (11) and (9), we get the KKT conditions

$$\mathbf{y}_t - \mathbf{X}_t\beta_t^* = \lambda\theta_t^*, t = 1, ..., T$$
$$M(\theta^*(\lambda)) = (X_1^T\theta_1^*, ..., X_T^T\theta_T^*) \in \partial\phi(\mathbf{B}^*).$$

The proof is complete. □

Since $\partial\phi(0)$ is a convex set, the feasible set $\mathcal{F}$ is also convex. Theorem 1 tells us that the solution $\theta^*(\lambda)$ of the dual problem (4) is the projection of $\mathbf{y}/\lambda$ onto the convex set $\mathcal{F}$. This geometric property plays a fundamentally important role in developing our screening rule.

4.2 Preliminary Screening Rules by KKT Condition

We first present the basic idea for developing our preliminary screening rules.

Denote $\phi_j^i(\mathbf{B}) = ||\mathbf{B}_{G_j^i}||_F$, and $\mathcal{B}_j^i = \{\zeta \in \mathcal{H} : ||\zeta|| \leq w_j^i\}$. Then the sub-differential of $\phi(\mathbf{B})$ becomes

$$\partial\phi(\mathbf{B}) = \sum_{i=1}^{d}\sum_{j=1}^{n_i} w_j^i \partial\phi_j^i(\mathbf{B}) \subseteq \sum_{i=1}^{d}\sum_{j=1}^{n_i}\mathcal{B}_j^i.$$

To be specific, for any node G_j^i, we have

$$\partial\phi_j^i(\mathbf{B}^*(\lambda)) = \begin{cases} \{\zeta \in \mathcal{H}_{G_j^i} : ||\zeta||_F \leq 1\}, \text{ if } [\mathbf{B}^*(\lambda)]_{G_j^i} = 0 \\ [\mathbf{B}^*(\lambda)]_{G_j^i} / ||[\mathbf{B}^*(\lambda)]_{G_j^i}||_F, \text{ otherwise} \end{cases}$$

This implies that

$$[\mathbf{B}^*(\lambda)]_{G_j^i} = 0 \text{ if } ||\partial\phi_j^i(\mathbf{B}^*(\lambda))|| < 1. \tag{12}$$

Since the KKT condition (6) in Theorem 1 indicates that $M(\theta^*(\lambda)) \in \partial\phi(\mathbf{B}^*)$, we can split $M(\theta^*(\lambda))$ as:

$$M(\theta^*(\lambda)) = (\mathbf{X}_1^T\theta_1^*, ..., \mathbf{X}_T^T\theta_T^*) = \sum_{i=0}^{d}\sum_{j=1}^{n_i}\xi_j^i, \tag{13}$$

where $\xi_j^i \in w_j^i\partial\phi_j^i(\mathbf{B}^*(\lambda)), \forall i \in 0 \cup [d], j \in [n_i]$. Combined with (12), it leads to the preliminary screening rule below, which is the key idea of the final screening method.

$$\forall i \in 0 \cup [d], j \in [n_i], [\mathbf{B}^*(\lambda)]_{G_j^i} = 0, \text{ if } ||\xi_j^i||_F < w_j^i, \quad (R^*)$$

This rule tells us that if the component ξ_j^i in $\partial\phi(\mathbf{B}^*)$ satisfies the condition that $||\xi_j^i||_F < w_j^i$, then the features included in node G_j^i are inactive.

Then, in order to split $M(\theta^*(\lambda))$ into the desired form (13), we adopt the efficient Hierarchical Projection algorithm (Algorithm 1) proposed in [17]. There is a slightly difference between Algorithm 1 and the original algorithm, that is, our algorithm is used to project a matrix instead of a vector.

Algorithm 1 Hierarchical Projection: $\mathbf{P}_{\mathcal{A}_1^0}(\cdot)$

```
1: Input: $\mathbf{Z} \in \mathcal{H}$, the index tree $T$ and positive weights $w_j^i$
   for all nodes $G_j^i$ in $T$.
2: Output: $\mathbf{U}^0 = \mathbf{P}_{\mathcal{A}_1^0}(\mathbf{Z}), \mathbf{V}^i$ for $\forall i \in 0 \cup [d]$.
3: Set $\mathbf{U}^i \leftarrow 0 \in \mathcal{H}, \mathbf{V}^i \leftarrow 0 \in \mathcal{H}, \forall i \in 0 \cup [d+1]$.
4: for $i = d$ to $0$ do
5:    for $j = 1$ to $n_i$ do
6:       $\mathbf{V}_{G_j^i}^i = \mathbf{P}_{\mathcal{B}_j^i}(\mathbf{Z}_{G_j^i} - \mathbf{U}_{G_j^i}^{i+1}), \mathbf{U}_{G_j^i}^i \leftarrow \mathbf{U}_{G_j^i}^{i+1} + \mathbf{V}_{G_j^i}^i$.
7:    end for
8: end for
```

Theorem 2. *Let $\mathbf{V}^i, i \in 0 \cup [d]$ be the output of Algorithm 1 with input $M(\theta^*)$, and $\{\xi_j^i : i \in 0 \cup [d], j \in [n_i]\}$ be the set of vectors that satisfy (13). Then*

$$\text{(i)}: \mathbf{V}_{\mathrm{G_j^i}}^{\mathrm{i}} \in \mathrm{w_j^i}\partial\phi_{\mathrm{j}}^{\mathrm{i}}(\mathbf{B}^*(\lambda)), \forall \mathrm{i} \in 0 \cup [\mathrm{d}], \mathrm{j} \in [\mathrm{n_i}].$$

$$\text{(ii)}: \mathbf{U}_{\mathrm{G_j^i}}^{\mathrm{i+1}} = \mathbf{P}_{\mathcal{C}_{\mathrm{j}}^{\mathrm{i}}}(\mathbf{Z}_{\mathrm{G_j^i}}), \textit{for any non-leaf node } \mathrm{G_j^i}.$$

According to part (i) of Theorem 2 and the preliminary rule (R^*), we have

$$||\mathbf{V}_{G_j^i}^i||_F < w_j^i \Rightarrow [B^*(\lambda)]_{G_j^i} = 0$$

By using part (ii) of Theorem 2 and the initialization that $\mathbf{U}_{G_j^i}^{i+1} = 0$ when G_j^i is a leaf node (step 3 in Algorithm 1), we finally get the following screening rule, that is $[\mathbf{B}^*(\lambda)]_{G_j^i} = 0$ when:

$$(a): ||\mathbf{P}_{\mathcal{B}_j^i}\big([M(\theta^*(\lambda))]_{G_j^i} - \mathbf{P}_{\mathcal{C}_j^i}\big([M(\theta^*(\lambda))]_{G_j^i}\big)\big)||_F < w_j^i,$$
$$\text{if } G_j^i \text{ is a non-leaf node.}$$

$$(b): ||\mathbf{P}_{\mathcal{B}_j^i}\big([M(\theta^*(\lambda))]_{G_j^i}\big)||_F < w_j^i, \text{ if } G_j^i \text{ is a leaf node.}$$

From the definition of $\mathbf{P}_{\mathcal{B}_j^i}(\cdot)$, we can further simplify them into the following form: $[\mathbf{B}^*(\lambda)]_{G_j^i} = 0$ if:

$$(a): ||[M(\theta^*(\lambda))]_{G_j^i} - \mathbf{P}_{\mathcal{C}_j^i}\big([M(\theta^*(\lambda))]_{G_j^i}\big)||_F < w_j^i, \quad \text{(R1}'\text{)}$$
$$\text{if } G_j^i \text{ is a non-leaf node.}$$

$$(b): ||[M(\theta^*(\lambda))]_{G_j^i}||_F < w_j^i, \text{ if } G_j^i \text{ is a leaf node.} \quad \text{(R2}'\text{)}$$

4.3 Estimation of the Dual Optimum

However, we can not apply the screening rules (R1′) and (R2′) in real applications directly, since the optimal solution $\theta^*(\lambda)$ is always unknown. Fortunately, we can estimate the region where the dual solution $\theta^*(\lambda)$ lies, if we are given one $\theta^*(\lambda_0)$ with $\lambda < \lambda_0$. Thus we need to know a $\theta^*(\lambda_0)$ at a special λ_0 as a starter. Next, we show how to find this special starter and how to estimate an appropriate region where $\theta^*(\lambda)$ lies in.

A special case for the dual solution is the one corresponding to the primal solution $\mathbf{B}^*(\lambda_0) = 0$. This can be reached by choosing a large enough λ_0. The following theorem enables us to find the smallest λ_0 that can make $\mathbf{B}^*(\lambda_0) = 0$.

Theorem 3. *For STM, we let $\lambda_{\max} = \max\{\lambda : \boldsymbol{y}/\lambda \in \mathcal{F}\}$ and define the operator $\mathbf{S}_1^0(\mathbf{Z}) = \mathbf{Z} - \mathbf{P}_{\mathcal{C}_1^0}(\mathbf{Z})$. Then,*

$$\text{(i)}: \lambda_{\max} = \{\lambda : ||\mathbf{S}_1^0((\mathbf{X}_1^{\mathrm{T}}\boldsymbol{y}_1/\lambda, ..., \mathbf{X}_{\mathrm{T}}^{\mathrm{T}}\boldsymbol{y}_{\mathrm{T}}/\lambda))||_{\mathrm{F}} = \mathrm{w}_1^0\}$$

$$\text{(ii)}: \frac{\boldsymbol{y}}{\lambda} \in \mathcal{F} \Leftrightarrow \lambda \geq \lambda_{\max} \Leftrightarrow \theta^*(\lambda) = \frac{\boldsymbol{y}}{\lambda} \Leftrightarrow \beta^*\lambda = 0.$$

We first introduce a useful tool to characterize the projection operators and provide the definition of the normal cone [17], [29] as:

Definition 2. *(Normal Cone) Let Ω be a nonempty convex subset of $\mathbb{R}^n$. Then the* ***normal cone*** *to the set Ω at $\boldsymbol{x} \in \Omega$ is defined by*

$$N_\Omega(\boldsymbol{x}) = \{\boldsymbol{u} \in \mathbb{R}^n | \langle \boldsymbol{u}, \boldsymbol{y} - \boldsymbol{x} \rangle \leq 0, \forall \boldsymbol{y} \in \Omega\}. \tag{14}$$

In the case where $\boldsymbol{x} \notin \Omega$, we define $N_\Omega(\boldsymbol{x}) = \emptyset$.

For estimating $\theta^*(\lambda)$, by utilizing the geometric properties of the dual problem (4) and the known solution $\theta^*(\lambda_0)$, we can bound the region in which the optimal dual solution $\theta^*(\lambda)$ may lie in a small ball Θ in $\mathbb{R}^N$. This is justified in the following theorem.

Theorem 4. *For the primal problem (3), suppose that $\theta^*(\lambda_0)$ is already known with $\lambda_0 \leq \lambda_{\max}$. For $\lambda \in (0, \lambda_0)$, we define*

$$\boldsymbol{n} = \begin{cases} \frac{\boldsymbol{y}}{\lambda_0} - \theta^*(\lambda_0), & \textit{if } \lambda_0 < \lambda_{\max} \\ (\mathbf{X}_1\mathbf{S}_1^0(\mathbf{X}_1^T\frac{\boldsymbol{y}_1}{\lambda_{\max}}); ...; \mathbf{X}_T\mathbf{S}_1^0(\mathbf{X}_T^T\frac{\boldsymbol{y}_T}{\lambda_{\max}})), & \textit{if } \lambda_0 = \lambda_{\max} \end{cases}$$

$$\boldsymbol{r}(\lambda, \lambda_0) = \frac{\boldsymbol{y}}{\lambda} - \theta^*(\lambda_0),$$

$$\boldsymbol{r}^\perp(\lambda, \lambda_0) = \boldsymbol{r}(\lambda, \lambda_0) - \frac{\langle \boldsymbol{r}(\lambda, \lambda_0), \boldsymbol{n}(\lambda_0)\rangle}{||\boldsymbol{n}(\lambda_0)||^2}\boldsymbol{n}(\lambda_0).$$

Then, the followings hold:
(i) $\boldsymbol{n} \in \mathrm{N}_{\mathcal{F}}(\theta^*(\lambda_0))$,
(ii) $||\theta^*(\lambda) - (\theta^*(\lambda_0) + \frac{1}{2}\boldsymbol{r}^\perp(\lambda, \lambda_0))|| \leq ||\frac{1}{2}\boldsymbol{r}^\perp(\lambda, \lambda_0)||$.

Theorem 4 indicates that $\theta^*(\lambda)$ is in a ball centered at $\mathbf{o}(\lambda, \lambda) = \theta^*(\lambda_0) + \frac{1}{2}\mathbf{r}^\perp(\lambda, \lambda_0)$ of radius $\frac{1}{2}||\mathbf{r}^\perp(\lambda, \lambda_0)||$, i.e.,

$$\Theta = \{\theta : ||\theta - \mathbf{o}(\lambda, \lambda_0)|| \leq \frac{1}{2}||\mathbf{r}^\perp(\lambda, \lambda_0)||\}. \tag{15}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2816029, IEEE Transactions on Knowledge and Data Engineering

4.4 The Proposed Screening Rules

In this section, based on the preliminary screening rules (R1′) and (R2′), we will develop screening rules for the case when we are given the ball Θ that $\theta^*(\lambda)$ lies in instead of the optimal solution $\theta^*(\lambda)$, which is always unknown in real word applications.

For each node G_j^i in the index tree and the ball Θ defined in (15) that $\theta^*(\lambda) \in \Theta$, we denote $[M(\Theta)]_{G_j^i} = \{[M(\theta)]_{G_j^i} : \theta \in \Theta\}$ and define an operator $\mathbf{S}_j^i(\mathbf{Z}) = \mathbf{Z}_{G_j^i} - \mathbf{P}_{\mathcal{C}_j^i}(\mathbf{Z}_{G_j^i})$, then (R1′) and (R2′) can be relaxed into:

$$(a) : s_j^i(\lambda, \lambda_0) = \sup_{\zeta} \{||\mathbf{S}_j^i(\zeta)|| : \zeta \in \Xi_j^i \supseteq [M(\Theta)]_{G_j^i}\} < w_j^i$$

$$\Rightarrow [\mathbf{B}^*(\lambda)]_{G_j^i} = 0 \text{ if } G_j^i \text{ is a non-leaf node,} \qquad (R1^*)$$

$$(b) : s_j^i(\lambda, \lambda_0) = \sup_{\zeta} \{||\zeta|| : \zeta \in \Xi_j^i\} < w_j^i$$

$$\Rightarrow [\mathbf{B}^*(\lambda)]_{G_j^i} = 0, \text{ if } G_j^i \text{ is a leaf node.} \qquad (R2^*)$$

where $\Xi_j^i = \{\zeta : \zeta \in \mathcal{H}_{G_j^i} \text{ and } ||\zeta - [M(\mathbf{o}(\lambda, \lambda_0))]_{G_j^i}|| \leq \frac{1}{2}||\mathbf{r}^{\perp}(\lambda, \lambda_0)|| \max_{1 \leq t \leq T}\{||\mathbf{X}_{G_j^i}^t||_2\}\}$.

In the above screening rules, we need to solve the following two optimization problems:

$$s_j^i(\lambda, \lambda_0) = \sup_{\zeta} \{||\mathbf{S}_j^i(\zeta)|| : \zeta \in \Xi_j^i\}, \text{ if } G_j^i \text{ is a non-leaf node,} \qquad (16)$$

$$s_j^i(\lambda, \lambda_0) = \sup_{\zeta} \{||\zeta|| : \zeta \in \Xi_j^i\}, \text{ if } G_j^i \text{ is a leaf node.} \qquad (17)$$

They are two non-convex problems since we need to find a supreme value for a convex objective function. We can show [17] that both of them enjoy close form solutions, as presented in the following two theorems, respectively.

Before giving the theorems, we need to firstly introduce the definitions of virtual node of an index tree, relative interior and relative boundary.

Definition 3. *(Virtual Node) [17] For a non-leaf node G_j^i of an index tree T, let $\mathcal{I}_c(G_j^i) = \{k : G_k^{i+1} \subset G_j^i\}$. If $G_j^i \setminus \cup_{k \in \mathcal{I}_c(G_j^i)} G_k^{i+1} \neq \emptyset$, we define a virtual node of G_j^i by $G_{j'}^{i+1} = G_j^i \setminus \cup_{k \in \mathcal{I}_c(G_j^i)} G_k^{i+1}$ for $j' \in \{n_{i+1}+1, n_{i+1}+2, ..., n_{i+1}+n'_{i+1}\}$, where n'_{i+1} is the number of virtual nodes of depth $i+1$.*

Definition 4. *(Relative Interior) Let $\mathcal{C}$ be a nonempty convex subset of $\mathbb{R}^n$. Then we say that $\boldsymbol{x}$ is a relative interior point of $\mathcal{C}$ if $\boldsymbol{x} \in \mathcal{C}$ and there exists an open sphere $\mathcal{S}$ centered at $\boldsymbol{x}$ such that $\mathcal{S} \cap \mathbf{aff}(\mathcal{C}) \subset \mathcal{C}$, i.e. $\boldsymbol{x}$ is an interior point of $\mathcal{C}$ relative to the affine hull of $\mathcal{C}$. All of relative interior points of $\mathcal{C}$ is called the relative interior of $\mathcal{C}$, and is denoted by $\mathbf{ri}(\mathcal{C})$.*

Definition 5. *(Relative Boundary) The relative boundary of a nonempty convex set $\mathcal{C}$ is defined as $\mathbf{rbd}(\mathcal{C}) = \mathbf{cl}(\mathcal{C})/\mathbf{ri}(\mathcal{C})$, where $\mathbf{cl}(\mathcal{C})$ and $\mathbf{ri}(\mathcal{C})$ are the closure and relative interior of $\mathcal{C}$.*

Lemma 2. *[24] For any non-root node G_j^i, we can find a unique path from G_j^i to the root G_1^0. Let the nodes on this path be $G_{r_l}^l$, where $l \in 0 \cup [i], r_0 = 1$, and $r_i = j$. Then,*

$$G_j^i \subset G_{r_l}^l, \forall l \in 0 \cup [i-1].$$
$$G_j^i \cap G_r^l = \emptyset, \forall r \neq r_l, l \in [i-1], r \in [n_i].$$

Theorem 5. *Let $\gamma = \frac{1}{2}||\boldsymbol{r}^{\perp}(\lambda, \lambda_0)|| \max_{1 \leq t \leq T}\{||\mathbf{X}_{G_j^i}^t||_2\}$ and $\boldsymbol{c} = [M(\boldsymbol{o}(\lambda, \lambda_0))]_{G_j^i}$and $\boldsymbol{v}^i, i \in 0 \cup [d]$ be the output of Algorithm 1 with input $M(\boldsymbol{o}(\lambda, \lambda_0))$. Then for problem (16), we have:*

- *Suppose that $\boldsymbol{c} \notin \mathcal{C}_j^i$. Then, $s_j^i(\lambda, \lambda_0) = ||\boldsymbol{v}_{G_j^i}^i|| + \gamma$.*
- *Suppose that node G_j^i has a virtual child node. Then, for any $\boldsymbol{c} \in \mathcal{C}_j^i$, $s_j^i(\lambda, \lambda_0) = \gamma$.*
- *Suppose that node G_j^i has no virtual child node. Then, the followings hold:*
  1) If $\boldsymbol{c} \in \mathbf{rbd}(\mathcal{C}_j^i)$, then $s_j^i(\lambda, \lambda_0) = \gamma$.
  2) If $\boldsymbol{c} \in \mathbf{ri}(\mathcal{C}_j^i)$, then for any node $G_k^t \subset G_j^i$, where $t \in \{i+1, ..., d\}$ and $k \in [n_t + n'_t]$, let the nodes on the path from G_k^t to G_j^i be $G_{r_l}^l$, where $l = i, ..., t, r_i = j$, and $r_t = k$, and $\Gamma(G_{r_{i+1}}^{i+1}, G_k^t) = \sum_{l=i+1}^{t}(w_{r_l}^l - ||\boldsymbol{v}_{G_{r_l}^l}^l||)$. Then, $s_j^i(\lambda, \lambda_0) = \left(\gamma - \min_{\{(k,t):G_k^t \subset G_j^i\}} \Gamma(G_{r_{i+1}}^{i+1}, G_k^t)\right)_+$.

Theorem 6. *For problem (17), we have $s_j^i(\lambda, \lambda_0) = ||[\mathbf{X}^T \boldsymbol{o}(\lambda, \lambda_0)]_{G_j^i}|| + \frac{1}{2}||\boldsymbol{r}^{\perp}(\lambda, \lambda_0)|| \max_{1 \leq t \leq T}\{||\mathbf{X}_{G_j^i}^t||_2\}$.*

Thus the screening rules $(R1^*)$ and $(R2^*)$ are developed.

4.5 Solving the Primal Problem of STM

We now turn to solving the optimization problem (3) in STM using our developed screening rules.

Just as most existing learning models, the optimal parameter value λ for some specific real application is always unknown. Typically, researchers adopt cross validation and stability selection to determine an appropriate parameter value. In each method, we need to solve STM many times along a grid of parameter values. It would be very time consuming. To this end, we solve a sequence of STM problems with the values of λ listed in a descending order. The detailed steps are given in Algorithm 2. Its main advantage is that the difference between two neighboring λ enable us to estimate θ_λ much more accurately, which leads to better performance in screening and finally improves the efficiency.

Algorithm 2 Screening for STM

```
1: Input: $\lambda_{\max} = \lambda_0 > \lambda_1 > ... > \lambda_{\mathcal{K}} = \lambda$, $\mathbf{B}^*(\lambda_0) = 0$.
2: Output: $\mathbf{B}^*(\lambda_{\mathcal{K}})$
3: for $k = 0$ to $\mathcal{K} - 1$ do
4:     Compute $\theta^*(\lambda_k)$ using Eq. (5)
5:     Estimate $\theta^*(\lambda_{k+1})$
6:     Do feature reduction using the screening rules $(R1^*)$
       and $(R2^*)$
7:     Compute $\mathbf{B}^*(\lambda_{k+1})$
8: end for
```

In the initialization step, we first find the parameter $\lambda_{\max}$ which makes $\mathbf{B}^*(\lambda_{\max}) = 0$ according to Theorem 3. And then we insert $\mathcal{K}+1$ parameters $\lambda_k (k = 0, ..., \mathcal{K})$ between $\lambda_{\max}$ and λ with $\lambda_0 = \lambda_{\max}$ and $\lambda_{\mathcal{K}} = \lambda$. Thus we get $\mathcal{K}+1$ STM problems and solve them one by one until we get $\mathbf{B}^*(\lambda)$ as the steps 4-6 listed in Algorithm 2.

5 Experiments

We evaluate the proposed STM and HFS on both synthetic and real data sets in terms of the performance of feature

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2816029, IEEE Transactions on Knowledge and Data Engineering

selection and efficiency. We compare our method with the following state-of-the-arts:

- Lasso [13]: Lasso equipped with screening for identifying genetic risk factors for AD.
- MT [23]: Multi-Task Learning with screening.
- TGL [16]: TGL equipped with screening for identifying genetic risk factors for AD.
- AFL [15]: AFL adopted non-convex model for identifying genetic risk factors for AD.

We perform all the computations on a single core of Intel(R) Core(TM) i7-5930K 3.50GHz, 128GB MEM.

5.1 Experiment on Synthetic Data

We perform experiments on two synthetic datasets. We use the smaller one, named synthetic 1, to visualize the performance of sparse pattern recovery by Lasso, MT, T-GL, AFL and STM. We use the larger one, named synthetic 2, to evaluate the efficiency of STM combined with the proposed HFS screening. For both synthetic 1 and synthetic 2, we create eight tasks and the true model is: $\mathbf{y}_t = \mathbf{X}_t\beta_t^* + 0.01\epsilon$, where $\mathbf{X}_t$ is sampled from a standard Gaussian, $\epsilon \sim N(0,1), t = 1,..,8$. For synthetic 1, we set $N_t = 20, p = 100$, and create a tree with three layers, where $n_1 = 5$, $n_2 = 10$, and $n_3 = 100$. For synthetic 2, we set $N_t = 600, p = 20000$ and build a tree with three layers, where $n_1 = 1000$, $n_2 = 2000$, and $n_3 = 20000$. The nodes in the same layer contain roughly the same number of features. To construct β_t^*, we first select 20% of the nodes in first layer and then 50% of their child nodes in layer 2. Then, the components of β_t^* corresponding to the selected nodes are sampled from a standard Gaussian. Thus 10% of the features have nonzero coefficients.

5.1.1 Tree-Structured Sparse Pattern Recovery

We employ stability selection to recover the tree structured sparse patterns of β_t^* for $t = 1,\ldots,8$. Specifically, for each trial, we randomly sample half of the data and solve a sequence of problems with λ equally spaced on the logarithmic scale of $\lambda/\lambda_{\max}$ from 1.0 to 0.05. We run 10 trials and report the averaged results.

Figure 3 shows the ground truth of $\mathbf{B} = (\beta_1^*,\ldots,\beta_8^*)$ and the recovery results by Lasso, MT, TGL, AFL and our proposed STM. The results in Figure 3 indicate that our method can exactly recover the tree-structured sparse patterns and significantly outperforms the competitors.

To give a quantitative comparison, we view the process of feature selection as a classification task. Specifically, we label 1 for the features with a nonzero coefficient and zero for others in the ground-truth. Figures 4 and 5 show the sensitivity and specificity—that are commonly used to measure the classification performance—of Lasso, MT, TGL, AFL and STM. We can see again that STM performs the best.

5.1.2 Evaluation of HFS

To evaluate the performance of the proposed HFS rule, we employ two metrics: speedup and rejection ratios. Specifically, speedup is the ratio of the running time of the solver without screening to the running time of solver with screening; rejection ratio for the i-th layer is defined by

$$r_i = \frac{\sum_{k\in\mathcal{G}^i}|G_k^i|}{p_0},$$

where p_0 is the number of zero coefficients in the solution matrix, $\mathcal{G}^i$ is the index set of the inactive nodes in depth i identified by our screening rules and $|G_k^i|$ is the number of

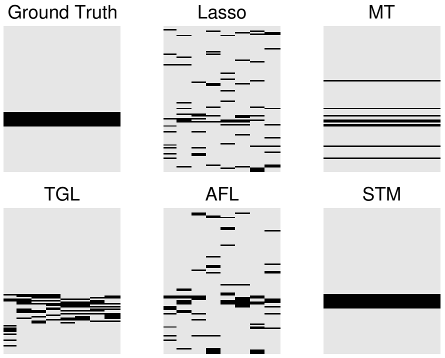


Fig. 3. Comparison results of selected features in synthetic 1. The horizontal axis is the eight tasks and the vertical axis is the features. Black pixels are the selected features.

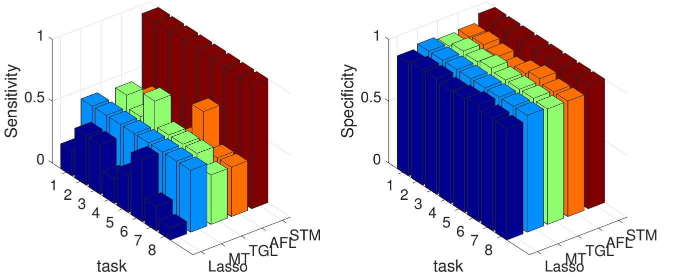


Fig. 4. The specificities and sensitivities of different models in each task on synthetic 1.

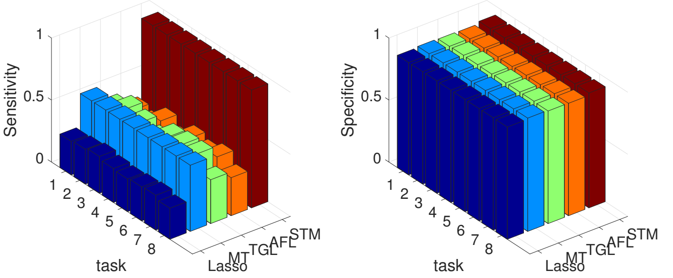


Fig. 5. The specificities and sensitivities of different models in each task on synthetic 2.

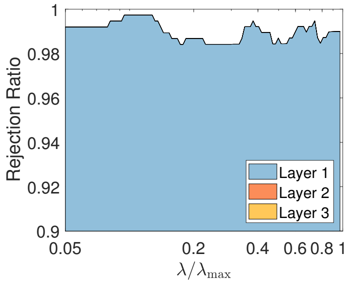


Fig. 6. The rejection ratios of HFS in different layers on Synthetic 2.

features contained in node G_k^i. The results are presented in Table 1 and Figure 6.

Table 1 shows the detailed running time of the solver with and without screening at a sequence of 100 values of λ on synthetic 2. We can see that HFS leads to a significant speedup—that is, 16.7 times on average. From the rejection ratio in Figure 6, we can see that HFS can identify more than 98% of the inactive nodes in the first layer.

TABLE 1
Running time (in seconds) for solving STM on synthetic 2 and ADNI. The second column: the solver without screening, the fourth column: the solver with screening, the last column: speedup.

Dataset	solver	HFS	HFS+solver	speedup
synthetic 2	3395.7	13.5	203.7	16.7
ADNI	18010.4	184.7	242.1	74.4

5.2 Experiment on ADNI Data

In this section, we conduct a series of experiments on Alzheimers Disease Neuroimaging Initiative (ADNI) dataset to identify the genetic risk factors for AD. Specifically, brain atrophy is usually used as a biomarker of AD in existing ADNI studies like [1]. Brain MRI is an effective tool to detect and measure brain atrophy. Therefore, in this experiment, we use the SNPs on Chr19 as the features and the volumes of several different brain regions as the responses to design an STM model to identify the genetic risk factors for AD. The concrete processes are shown below.

5.2.1 Data Preprocessing

The ADNI dataset has a total of 1,319 subjects, comprised of 327 healthy controls, 249 AD patients, 41 participants with Mild Cognitive Impairment (MCI) [1], [4], 220 early MCI (EMCI) patients, 419 late MCI (LMCI) patients, and 63 patients with Significant Memory Concerns (SMC) [30]. In this study, the T1 magnetic resonance imaging (MRI) [4] volumes of the major influenced brain regions, such as the hippocampus (HIPP) and the entorhinal cortex (EC) [1], which are closely related to Alzheimers disease, are chosen as a part of features. Therefore, each subject is represented by one SNPs data and one T1 MRI representation, which is extracted from the original volumes by Freesurfer [31].

For SNPs data, we perform standard quality control in PLINK [32], which is a tool set and popularly used to analyze the whole-genome association and population-based linkage relationship, as data preprocessing. Firstly, we remove SNPs whose minor allele frequency [33] $< 5\%$, or missingness $> 5\%$, or the deviations from Hardy-Weinberg Equilibrium P [34] $< 5 \times 10^{-7}$.Then we delete the subjects from the dataset, whose missing SNPs are more than 10%. After this, we impute the data using a well known method called MaCH [35], [36], which is a Markov chain-based framework for genotype imputation and haplotyping. MaCH is adopted to estimate haplotypes and inference of missing genotypes by sequence and genotype data. In addition, we apply several filter rules to the imputed data, including RSQ (estimated R^2 specific to each SNP) > 0.5, FREQ1 (frequency for reference Allele 1) $> 1\%$ and FREQ1 $< 99\%$. Finally, we get the training data with 1,319 subjects and each subject has 155,357 SNPs from chromosome 19.

For the MRI data of each subject, we measure the volumes of 4 different regions—left entorhinal cortex (LEH), left hippocampus (LHP), right entorhinal cortex (REH), and right hippocampus (RHP)—in the brain as the responses for that subject. Thus, we can construct 4 responses for each subject from its MRI data.

5.2.2 Tasks in Our Model

We construct 4 related tasks using the features matrix and the 4 responses in the preprocessed dataset. For the t-th task, the feature matrix is denoted as $\mathbf{X}_t \in \mathbb{R}^{1,319 \times 155,357}$ and each row presents a subject, each column is a SNP. The response $\mathbf{y}_t$ for this task is a vector in $\mathbb{R}^{155,357}$ and the component $\mathbf{y}_t(i)$ is the volume of the t-th brain region of the i-th subject. Our goal is to identify the genetic risk factors for AD by studying the associations between the feature matrix X_t and the response vector $\mathbf{y}_t$.

5.2.3 Construct the Tree Structure for Features

We build a four layer index tree for grouping the SNPs according to their pairwise R^2 values and loci. In general, since the pairwise R^2 is unknown, we first construct an index tree for a reference dataset that contains 42,183 SNPs on Chr19 from HapMap release #27 (http://hapmap.ncbi.nlm.nih.gov/downloads/ld_data/latest/ld_chr19_CEU.txt.gz), who has entire pairwise R^2 values. And then we map the index tree into our target dataset. The detail steps are described below.

We first align chromosomal loci and refSNP cluster ID in the reference dataset and the target dataset according to Genome Reference Consortium GRCh37 using UCSC LiftOver tool [37]. We group the adjacent SNPs together in the reference dataset if their pairwise R^2 is nonzero, which forms 1,230 nodes. This is the first layer of the index tree for the reference dataset. Then, for each node in the first layer, we set the threshold of R^2 to be 0.01. That is we groups the adjacent SNPs together if their pairwise R^2 is larger or equal to 0.01, which leads to the second layer of the index tree. After that, we increase the threshold to 0.1 and construct the third layer in the same way. Thus we obtain a three layers index tree for the reference dataset.

Then, we map the three layers index tree from the reference dataset to the target dataset. Specifically, for each node in the constructed tree starting from locus a and ending at locus b, we group the SNPs in the target dataset whose loci are between a and b into a node in the index tree for the target dataset. If there is no SNP whose locus is between a and b in the target dataset, we skip to the next node. In this way, we can obtain a three layers index tree for the target dataset. At last, we split each node in the last layer into nodes with a single SNPs, which forms the forth layer with 155,357 nodes. As a result, we get a tree with a root layer and additional layers 1 to 4. And the nodes numbers from layer 1 to layer 4 are 1,063, 5,133, 14,883 and 155,357 respectively.

5.2.4 Evaluation of Efficiency

In this experiment, we evaluate the efficiency of STM with and without HFS on the ADNI data set.

We solve a sequence of STM models with and without screening with different parameter λ, which equally spaced on the logarithmic scale of $\lambda/\lambda_{\max}$ from 1.0 to 0.5. The result in Table 1 shows that our screening method can speed up STM model 74.4 times in average. In addition, we notice that the speedup here is much significant than in the synthetic experiment. Thus, we also expect HFS performs better in high dimensional applications.

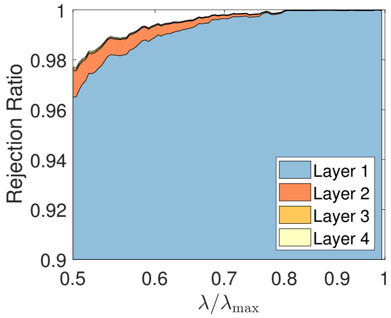


Fig. 7. The rejection ratios of HFS in different layers on ADNI data set.

The rejection ratio results in Figure 7 shows that most of the identified inactive nodes are screened in the fist layer. After two layers screening, almost all the inactive nodes, i.e. $\sum_{i=1}^{4} r_i > 98\%$ are identified.

5.2.5 Ranking Predictors via Stability Selection

In this subsection, we evaluate the performance of our model in identifying the genetic risk factors for AD.

We first randomly subsample half of the subjects in the preprocessed dataset for 100 times. Then on each sampled dataset, we solve the STM model with a sequence of λ equally spaced on the logarithmic scale of $\lambda/\lambda_{\max}$ from 1.0 to 0.5. After that, we use stability selection [38] to ranking the SNPs identified in the solutions. At last, to show the relevance of the top ranked SNPs with AD further, we calculate the p-values of the top 100 SNPs in each task. We repeat this process for the other two baseline methods and get the ranked SNPs and their p-values (Pearson correlation coefficient).

We show the p-values of the top 100 SNPs in 4 tasks against their positions in Chr19 for each model in Figure 8. All the four sub-figures show that the top 100 SNPs selected by MT are spread over almost the whole region of Chr19. This makes it difficult to tell which genes are more likely to be the disease-causing genes linked to AD. It appears that TGL performs better than MT, since its identified SNPs are distributed in a few chromosome regions. However, in the results with LEH and LHP as responses (upper left and lower left panels), TGL also identifies some SNPs in the first half of Chr19. Since there are no known disease-causing genes in that regions, it is more likely that they are irrelevant to AD. In contrast, the SNPs selected by our method spread in a much narrower region. We give a more detailed plot in Figure 9, and we can see that many of our selected SNPs are located in several genes such as APOE, PVRL2, TOMM40 and APOC1, which are all already repeatedly proved to be implicated in AD risk or risks for other neuropsychiatric disorders [4], [39], [40], [41], [42]. To be specific, AlzGene [39] gives ten candidate genes with high AD-risk, including LDLR, GAPDHS, BCAM, PVRL2, TOMM40, APOE, APOC1, APOC4, EXOC3L2, and CD33, whose positions are presented in Figure 10. These ten candidate genes have been considered as the most strongly associated genes with AD on Chr.19. Moreover, the study in [42] reveals that APOE is the major susceptibility gene for sporadic late-onset AD. In [4], [40], APOE and PVRL2 are proved to be significantly associated with AD. The researchers in [41] demonstrate the association between APOC1 and AD risk in Caucasians, Asians and Caribbean Hispanics.

TABLE 2
The p-values and the corresponding genes of top 30 SNPs identified by STM. Columns 1 to 6 represent RS_ID of selected SNPs, p-value of LEH, LHP, REH and RHP, and the corresponding gene.

RS_ID	LEH	LHP	REH	RHP	Gene
rs3745150	3e-04	4e-06	7e-07	1e-06	PVRL2
19:45386467	2e-04	3e-07	4e-07	1e-07	PVRL2
rs12972156	2e-04	3e-07	4e-07	1e-07	PVRL2
rs12972970	2e-04	3e-07	4e-07	1e-07	PVRL2
rs283810	2e-03	1e-05	8e-06	5e-06	PVRL2
rs283811	3e-04	6e-08	3e-07	4e-08	PVRL2
rs283812	5e-04	1e-06	3e-06	1e-06	PVRL2
rs283813	5e-01	2e-01	2e-02	7e-02	PVRL2
rs283814	5e-04	9e-07	3e-06	1e-06	PVRL2
rs283815	1e-04	1e-08	3e-08	2e-09	PVRL2
rs76692773	2e-04	2e-07	5e-07	7e-08	TOMM40
rs71352238	6e-04	6e-07	3e-06	5e-07	TOMM40
rs184017	9e-03	5e-04	5e-05	3e-05	TOMM40
rs2075649	2e-04	2e-07	4e-07	6e-08	TOMM40
rs2075650	5e-04	6e-07	2e-06	5e-07	TOMM40
rs157581	2e-03	1e-03	1e-05	4e-04	TOMM40
rs34095326	2e-04	2e-07	4e-07	7e-08	TOMM40
rs34404554	2e-04	2e-07	4e-07	7e-08	TOMM40
rs11556505	6e-04	6e-07	3e-06	5e-07	TOMM40
rs157582	2e-04	1e-07	1e-06	1e-07	TOMM40
19:45406538	6e-03	5e-05	2e-04	2e-06	APOE
rs7259620	2e-03	2e-04	3e-04	4e-03	APOE
rs769446	3e-02	3e-03	4e-02	5e-03	APOE
rs405509	1e-02	3e-03	1e-03	1e-03	APOE
rs440446	5e-07	4e-11	1e-08	3e-11	APOE
rs769450	2e-07	3e-12	9e-10	2e-12	APOE
rs1081106	2e-04	1e-06	6e-05	4e-06	none
rs445925	6e-07	4e-11	1e-08	3e-11	APOC1
rs10414043	6e-07	4e-11	1e-08	3e-11	APOC1
rs7256200	4e-05	4e-08	3e-06	1e-07	APOC1

We list the top 30 SNPs identified by our model together with their p-values in each task and the genes they belong to in Table 2. The important thing is that some of our top ranked SPNs are located in the gene APOE, PVRL2, TOMM40 and APOC1. As we discussed above, all of these genes have been proved to be associated with AD. This verifies that our model can identify the genetic risk factors for AD.

Table 3 lists the irrelevant SNPs wrongly selected by TGL. We can see that these selected SNPs located in four genes: ZNF682, VAV1, CD320 and SIGLECL1. From Figure 10, we can see that these genes have no confident relationships with AD.

Synthesizing the above experiments in Figures 8,9 and Table 2, we can see that our model STM has superiority over the baseline methods Lasso, AFL, TGL and MT. This should result from the fact that compared with TGL, our model can learn from multiple related tasks simultaneously, and that compared with MT, our method has an effective tree structure to group the SNPs according to their relationships.

At last, our work obtains the positive medical evaluation from Dr. Li Liu (homepage: https://biodesign.asu.edu/li-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2018.2816029, IEEE Transactions on Knowledge and Data Engineering

TABLE 3
The irrelevant SNPs in the top 100 SNPs selected by TGL. From Column 2 to Column 4, they represent RS_ID of selected SNPs, the corresponding p-value and gene, respectively.

Region	RS_ID	p_value	Gene
LEH	19:20115643	4e-03	ZNF682
	19:6852734	3e-03	VAV1
	19:6861043	2e-03	none
	19:6865229	3e-02	none
	19:6871492	8e-02	none
	rs10414931	7e-03	none
	rs112526880	3e-03	none
	rs11666078	7e-01	none
	rs11669968	2e-03	none
	rs11672182	1e-01	none
	rs11672913	2e-01	none
	rs146666686	2e-03	none
	rs17718517	2e-03	none
	rs186715	1e-01	none
	rs1990216	2e-03	none
	rs331689	1e-02	none
	rs331690	6e-03	none
	rs331691	1e-02	none
	rs461970	2e-03	VAV1
	rs55673918	1e-03	none
	rs55822857	4e-04	none
	rs56079027	1e-03	none
	rs56931055	1e-02	none
	rs58179654	2e-02	none
	rs59055395	3e-02	none
	rs60967546	4e-02	none
	rs61471228	2e-03	none
	rs7255262	4e-02	none
	rs72982472	1e-01	none
	rs8108918	8e-03	none
LHP	19:8369544	2e-03	CD320
	rs11668174	3e-03	SIGLECL1
	rs11672446	6e-03	SIGLECL1
	rs2002602	5e-02	CD320
	rs2232778	2e-03	CD320
	rs2913943	2e-02	CD320
	rs2927707	1e-02	CD320
	rs2927708	2e-03	CD320
	rs2927709	2e-03	CD320
	rs56303780	2e-03	CD320
	rs890856	8e-02	CD320

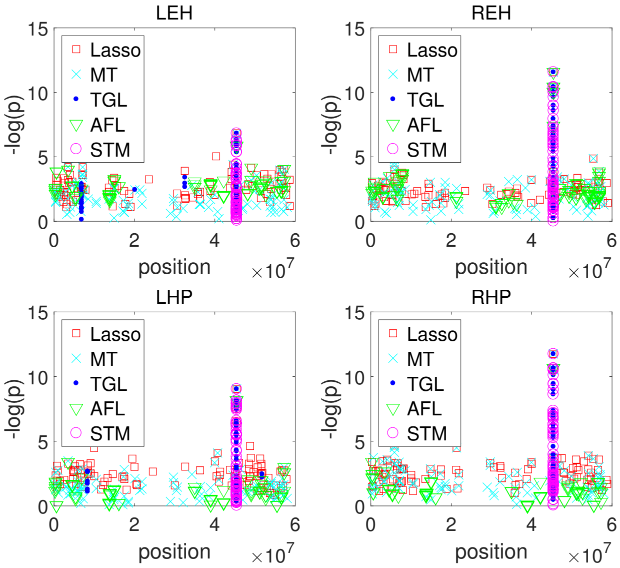


Fig. 8. The p values of the top 100 SNPs obtained by different methods. The horizontal axis is mapped to chromosome position.

liu), who is an assistant professor of Biomedical Informatics and a trained clinician. She is one of medical professionals with experience in AD and appreciates the critical roles genomic medicine and bioinformatics play in advancing precision medicine. Her medical evaluation is summarized as follows:

> Medical Evaluation: Progressive atrophy of the entorhinal cortex and the hippocampus is an early indicator of dementia even prior to clinically observable symptoms [43]. By analyzing brain MRI images and SNP data with the new STM method, we identified a narrow genomic region on chromosome 19 that was significantly associated with in vivo volume of four subregions of this area. This genomic region contains four genes (PVRL2, TOMM40, APOE and APOC1) with proved functions in the pathogenesis of AD [44]. In particular, we noticed that STM identified a common set of SNPs as genetic risk factors of atrophy in all of the four subregions (Table 2). Compared to other SNPs with inconsistent associations reported by other methods, these SNPs are more likely to contain causal variants (or at least linked to causal variants) that predispose an individual to AD. Fine-mapping causal variants is a grand challenge in G-WAS analysis. By pinpointing a set of candidate SNPs associated with multiple clinical responses, STM can greatly accelerate the discovery of causal variants for complex diseases.

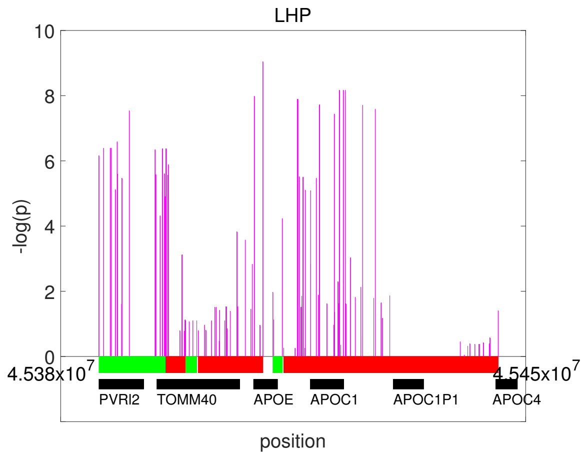


Fig. 9. The detailed plot of the p values of the top 100 SNPs obtained by STM with LHP as response. SNP groups in the layer 2 of our tree structure are plotted as green and red blocks on the chromosome. The genes distributed along the chromosome are plotted as black blocks.

6 Conclusion

In this paper, we propose a novel feature learning method called STM for identifying genetic risk factors for AD. The key idea is to explore in-depth the hierarchical structure among the features and the commonality across related tasks at the same time. Essentially, it is a multi-task feature learning model integrated with a tree structure for the features. The extensive experiment results on both synthetic and real datasets show that our model has great superiority over the existing methods in feature learning. In addition, to improve the time efficiency, we develop an effective screening rule called HFS for our method, which can improve the time-efficiency by several orders of magnitude. In the future, we plan to investigate models for finding the most related tasks from numerous tasks automatically.

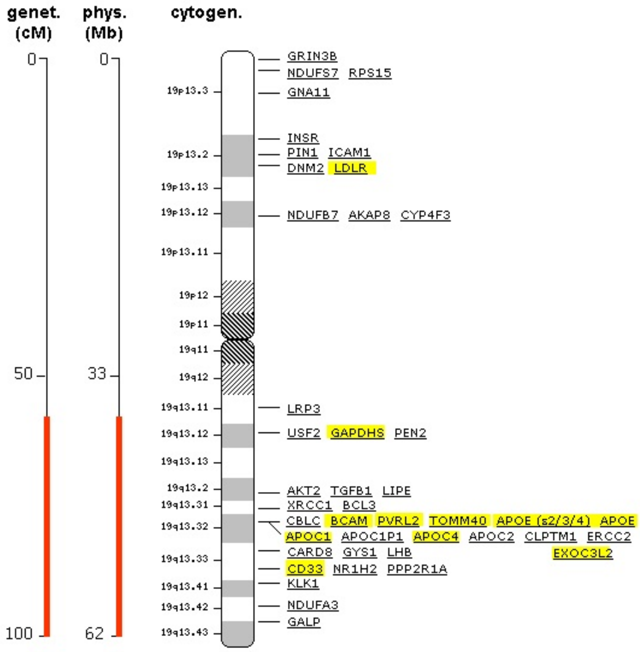


Fig. 10. AD-risk genes (marked by yellow) on Chr.19 according to AlzGene. Figure adapted from http://www.alzgene.org/chromo.asp?c=19.

ACKNOWLEDGMENTS

This work was supported by National Youth Top-notch Talent Support Program. The authors would like to specially thank Dr. Li Liu (liliu@asu.edu) for providing the professional medical evaluation.

REFERENCES

[1] L. Shen, S. Kim, S. L. Risacher, K. Nho, S. Swaminathan, J. D. West, T. Foroud, N. Pankratz, J. H. Moore, C. D. Sloan *et al.*, "Whole genome association study of brain-wide imaging phenotypes for identifying quantitative trait loci in mci and ad: A study of the adni cohort," *Neuroimage*, vol. 53, no. 3, pp. 1051–1063, 2010.

[2] M. Liu, D. Zhang, P.-T. Yap, and D. Shen, "Tree-guided sparse coding for brain disease classification," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2012, pp. 239–247.

[3] X. Hao, J. Yu, and D. Zhang, "Identifying genetic associations with mri-derived measures via tree-guided sparse learning," in *International Conference on Medical Image Computing and Computer-assisted Intervention*. Springer, 2014, pp. 757–764.

[4] N. Schuff, N. Woerner, L. Boreta, T. Kornfield, L. Shaw, J. Trojanowski, P. Thompson, C. Jack Jr, M. Weiner, and A. D. N. Initiative, "Mri of hippocampal volume loss in early alzheimer's disease in relation to apoe genotype and biomarkers," *Brain*, vol. 132, no. 4, pp. 1067–1077, 2009.

[5] R. Sladek, G. Rocheleau, J. Rung, C. Dina, L. Shen, D. Serre, P. Boutin, D. Vincent, A. Belisle, S. Hadjadj *et al.*, "A genome-wide association study identifies novel risk loci for type 2 diabetes," *Nature*, vol. 445, no. 7130, pp. 881–885, 2007.

[6] J. Flannick, G. Thorleifsson, N. L. Beer, S. B. Jacobs, N. Grarup, N. P. Burtt, A. Mahajan, C. Fuchsberger, G. Atzmon, R. Benediktsson *et al.*, "Loss-of-function mutations in slc30a8 protect against type 2 diabetes," *Nature genetics*, vol. 46, no. 4, pp. 357–363, 2014.

[7] B. D. Gelb and W. K. Chung, "Complex genetics and the etiology of human congenital heart disease," *Cold Spring Harbor perspectives in medicine*, vol. 4, no. 7, p. a013953, 2014.

[8] Y. Li, N. T. Klena, G. C. Gabriel, X. Liu, A. J. Kim, K. Lemke, Y. Chen, B. Chatterjee, W. Devine, R. R. Damerla *et al.*, "Global genetic analysis in mice unveils central role for cilia in congenital heart disease," *Nature*, vol. 521, no. 7553, pp. 520–524, 2015.

[9] M. A. Nalls, N. Pankratz, C. M. Lill, C. B. Do, D. G. Hernandez, M. Saad, A. L. DeStefano, E. Kara, J. Bras, M. Sharma *et al.*, "Large-scale meta-analysis of genome-wide association data identifies six new risk loci for parkinson's disease," *Nature genetics*, vol. 46, no. 9, pp. 989–993, 2014.

[10] A. Verstraeten, J. Theuns, and C. Van Broeckhoven, "Progress in unraveling the genetic etiology of parkinson disease in a genomic era," *Trends in Genetics*, vol. 31, no. 3, pp. 140–149, 2015.

[11] S. Przedborski, "The two-century journey of parkinson disease research," *Nature Reviews Neuroscience*, vol. 18, no. 4, pp. 251–259, 2017.

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

[13] T. Yang, J. Wang, Q. Sun, D. P. Hibar, N. Jahanshad, L. Liu, Y. Wang, L. Zhan, P. M. Thompson, and J. Ye, "Detecting genetic risk factors for alzheimer's disease in whole genome sequence data via lasso screening," in *Biomedical Imaging (ISBI), 2015 IEEE 12th International Symposium on*. IEEE, 2015, pp. 985–989.

[14] O. Kohannim, D. P. Hibar, J. L. Stein, N. Jahanshad, X. Hua, P. Rajagopalan, A. Toga, C. R. Jack Jr, M. W. Weiner, G. I. De Zubicaray *et al.*, "Discovery and replication of gene influences on brain structure using lasso regression," *Frontiers in neuroscience*, vol. 6, p. 115, 2012.

[15] T. Yang, J. Liu, P. Gong, R. Zhang, X. Shen, and J. Ye, "Absolute fused lasso and its application to genome-wide association studies," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*. ACM, 2016, pp. 1955–1964.

[16] Y. Li, J. W. Wang, T. Yang, J. Chen, C. Shen, L. Liu, L. Zhan, D. P. Hibar, N. Jahanshad, Y. Wang, S. D. Zhao, P. Thompson, and J. Ye, "Identification of alzheimer's disease risk factors by tree-structured group lasso screening," *The IEEE International Symposium on Biomedical Imaging*, 2016.

[17] J. Wang and J. Ye, "Multi-layer feature reduction for tree structured group lasso via hierarchical projection," in *Advances in Neural Information Processing Systems*, 2015, pp. 1279–1287.

[18] J. K. Pritchard and M. Przeworski, "Linkage disequilibrium in humans: models and data," *The American Journal of Human Genetics*, vol. 69, no. 1, pp. 1–14, 2001.

[19] T. Evgeniou, C. A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *Journal of Machine Learning Research*, vol. 6, no. Apr, pp. 615–637, 2005.

[20] J. Chen, L. Tang, J. Liu, and J. Ye, "A convex formulation for learning shared structures from multiple tasks," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 137–144.

[21] S. Kim and E. P. Xing, "Tree-guided group lasso for multi-task regression with structured sparsity," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*. Omnipress, 2010, pp. 543–550.

[22] J. Wang, J. Zhou, P. Wonka, and J. Ye, "Lasso screening rules via dual polytope projection," in *Advances in Neural Information Processing Systems*, 2013, pp. 1070–1078.

[23] J. Wang and J. Ye, "Safe screening for multi-task feature learning with multiple data matrices," in *International Conference on Machine Learning*, 2015, pp. 1747–1756.

[24] J. Liu and J. Ye, "Moreau-yosida regularization for grouped tree structure learning," in *Advances in Neural Information Processing Systems*, 2010, pp. 1459–1467.

[25] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *Advances in neural information processing systems*, 2007, pp. 41–48.

[26] J. Fan and J. Lv, "Sure independence screening for ultrahigh dimensional feature space," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 70, no. 5, pp. 849–911, 2008.

[27] L. E. Ghaoui, V. Viallon, and T. Rabbani, "Safe feature elimination for the lasso and sparse supervised learning problems," *arXiv preprint arXiv:1009.4219*, 2010.

[28] M. A. Hanson, "Invexity and the kuhn–tucker theorem," *Journal of mathematical analysis and applications*, vol. 236, no. 2, pp. 594–604, 1999.

[29] B. S. Mordukhovich and N. M. Nam, “Geometric approach to convex subdifferential calculus,” *Optimization*, vol. 66, no. 6, pp. 839–873, 2017.

[30] E. M. Reiman, K. Chen, G. E. Alexander, R. J. Caselli, D. Bandy, D. Osborne, A. M. Saunders, and J. Hardy, “Functional brain abnormalities in young adults at genetic risk for late-onset alzheimer’s dementia,” *Proceedings of the National Academy of Sciences*, vol. 101, no. 1, pp. 284–289, 2004.

[31] M. Reuter, N. J. Schmansky, H. D. Rosas, and B. Fischl, “Within-subject template estimation for unbiased longitudinal image analysis,” *Neuroimage*, vol. 61, no. 4, pp. 1402–1418, 2012.

[32] S. Purcell, B. Neale, K. Todd-Brown, L. Thomas, M. A. Ferreira, D. Bender, J. Maller, P. Sklar, P. I. De Bakker, M. J. Daly *et al.*, “Plink: a tool set for whole-genome association and population-based linkage analyses,” *The American Journal of Human Genetics*, vol. 81, no. 3, pp. 559–575, 2007.

[33] A. E. Shearer, R. W. Eppsteiner, K. T. Booth, S. S. Ephraim, J. Gurrola II, A. Simpson, E. A. Black-Ziegelbein, S. Joshi, H. Ravi, A. C. Giuffre *et al.*, “Utilizing ethnic-specific differences in minor allele frequency to recategorize reported pathogenic deafness variants,” *The American Journal of Human Genetics*, vol. 95, no. 4, pp. 445–453, 2014.

[34] S. Rodriguez, T. R. Gaunt, and I. N. Day, “Hardy-weinberg equilibrium testing of biological ascertainment for mendelian randomization studies,” *American journal of epidemiology*, vol. 169, no. 4, pp. 505–514, 2009.

[35] Y. Li, C. Willer, S. Sanna, and G. Abecasis, “Genotype imputation,” *Annual review of genomics and human genetics*, vol. 10, pp. 387–406, 2009.

[36] Y. Li, C. J. Willer, J. Ding, P. Scheet, and G. R. Abecasis, “Mach: using sequence and genotype data to estimate haplotypes and unobserved genotypes,” *Genetic epidemiology*, vol. 34, no. 8, pp. 816–834, 2010.

[37] R. M. Kuhn, D. Karolchik, A. S. Zweig, T. Wang, K. E. Smith, K. R. Rosenbloom, B. Rhead, B. J. Raney, A. Pohl, M. Pheasant *et al.*, “The ucsc genome browser database: update 2009,” *Nucleic acids research*, vol. 37, no. suppl_1, pp. D755–D761, 2008.

[38] N. Meinshausen and P. Bühlmann, “Stability selection,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 4, pp. 417–473, 2010.

[39] L. Bertram, M. B. McQueen, K. Mullin, D. Blacker, and R. E. Tanzi, “Systematic meta-analyses of alzheimer disease genetic association studies: the alzgene database,” *Nature genetics*, vol. 39, no. 1, pp. 17–23, 2007.

[40] M. W. Logue, M. Schu, B. N. Vardarajan, J. Buros, R. C. Green, R. C. Go, P. Griffith, T. O. Obisesan, R. Shatz, A. Borenstein *et al.*, “A comprehensive genetic association study of alzheimer disease in african americans,” *Archives of neurology*, vol. 68, no. 12, pp. 1569–1579, 2011.

[41] 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 alzheimer’s disease: a case-control study and meta-analysis,” *PloS one*, vol. 9, no. 1, p. e87017, 2014.

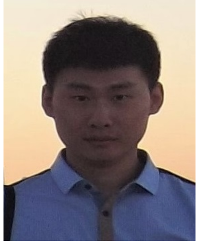
[42] K. D. Coon, A. J. Myers, D. W. Craig, J. A. Webster, J. V. Pearson, D. H. Lince, V. L. Zismann, T. G. Beach, D. Leung, L. Bryden *et al.*, “A high-density whole-genome association study reveals that apoe is the major susceptibility gene for sporadic late-onset alzheimer’s disease.” *The Journal of clinical psychiatry*, vol. 68, no. 4, pp. 613–618, 2007.

[43] G. B. Frisoni, N. C. Fox, C. R. Jack, P. Scheltens, and P. M. Thompson, “The clinical use of structural mri in alzheimer disease,” *Nature Reviews Neurology*, vol. 6, no. 2, pp. 67–77, 2010.

[44] R. J. Guerreiro, D. R. Gustafson, and J. Hardy, “The genetic architecture of alzheimer’s disease: Beyond app, psens and apoe,” *Neurobiology of aging*, vol. 33, no. 3, pp. 437–456, 2012.

Weizhong Zhang received the BS degree in Math and Applied Mathematics from Zhejiang University, China, in 2012. In 2017, he received the Ph.D degree in Computer Science also from Zhejiang University. He is currently a senior researcher in Tencent AI Lab, Shenzhen, China. His research interests include machine learning, computer vision, and data mining.

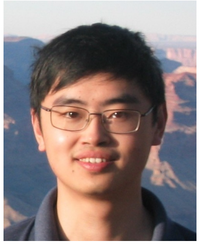
Tingjin Luo is a PhD candidate with the College of Science at the National University of Defense Technology, Changsha, China. He received his Master degree and B.S. degree with the College of Information System and Management at the same university in 2013 and 2011, respectively. His research interests include machine learning, data mining and computer vision.



Shuang Qiu is currently a PhD student of Computer Science and Engineering at the University of Michigan. His research interests include machine learning theory and its application to data mining.

Jieping Ye received the PhD degree in computer science from the University of Minnesota, Twin Cities, MN, in 2005. He is currently an associate professor of Electrical Engineering and Computer Science at University of Michigan, Ann Arbor, MI, USA. His research interests include machine learning, data mining, and biomedical informatics. He received the Outstanding Student Paper Award at ICML in 2004, the SCI Young Investigator of the Year Award at ASU in 2007, the SCI Researcher of the Year Award at ASU in 2009, the US National Science Foundation (NSF) CAREER Award in 2010, the KDD Best Research Paper Award honorable mention in 2010, and the KDD Best Research Paper Nomination in 2011 and 2012. He is a senior member of the IEEE.

Deng Cai is a Professor in the State Key Lab of CAD&CG, College of Computer Science at Zhejiang University, China. He received the Ph.D. degree in computer science from University of Illinois at Urbana Champaign in 2009. His research interests include machine learning, data mining and information retrieval.



Xiaofei He received the BS degree in Computer Science from Zhejiang University, China, in 2000 and the Ph.D. degree in Computer Science from the University of Chicago, in 2005. He is a Professor in the State Key Lab of CAD&CG at Zhejiang University, China. Prior to joining Zhejiang University, he was a Research Scientist at Yahoo! Research Labs, Burbank, CA. His research interests include machine learning, information retrieval and computer vision.

Jie Wang received the B.Sc. degree in electronic information science and technology from University of Science and Technology of China (USTC) in 2005, and the Ph.D. degree in computational science from the Florida State University in 2011. He is a full professor in the Department of Electronic Engineering and Information Science at USTC. His research interests include large scale optimization, machine learning, data mining, image processing, etc., and their applications to biomedical informatics.