FeaFiner: Biomarker Identification from Medical Data through Feature Generalization and Selection

Jiayu Zhou[1,2], Zhaosong Lu[3], Jimeng Sun[4], Lei Yuan[1,2], Fei Wang[4], Jieping Ye[1,2]
[1]Center for Evolutionary Medicine and Informatics, The Biodesign Institute, ASU, Tempe, AZ
[2]Department of Computer Science and Engineering, ASU, Tempe, AZ
[3]Department of Mathematics, Simon Fraser University, B.C., Canada
[4]IBM T.J. Watson Research Center, Yorktown Heights, NY

ABSTRACT

Traditionally, feature construction and feature selection are two important but separate processes in data mining. However, many real world applications require an integrated approach for creating, refining and selecting features. To address this problem, we propose *FeaFiner* (short for Feature Refiner), an efficient formulation that simultaneously generalizes low-level features into higher level concepts and then selects relevant concepts based on the target variable. Specifically, we formulate a double sparsity optimization problem that identifies groups in the low-level features, generalizes higher level features using the groups and performs feature selection. Since in many clinical researches non-overlapping groups are preferred for better interpretability, we further improve the formulation to generalize features using mutually exclusive feature groups. The proposed formulation is challenging to solve due to the orthogonality constraints, non-convexity objective and non-smoothness penalties. We apply a recently developed augmented Lagrangian method to solve this formulation in which each subproblem is solved by a non-monotone spectral projected gradient method. Our numerical experiments show that this approach is computationally efficient and also capable of producing solutions of high quality. We also present a generalization bound showing the consistency and the asymptotic behavior of the learning process of our proposed formulation.

Finally, the proposed FeaFiner method is validated on Alzheimer's Disease Neuroimaging Initiative dataset, where low-level biomarkers are automatically generalized into robust higher level concepts which are then selected for predicting the disease status measured by Mini Mental State Examination and Alzheimer's Disease Assessment Scale cognitive subscore. Compared to existing predictive modeling methods, FeaFiner provides intuitive and robust feature concepts and competitive predictive accuracy.

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*; J.3 [**Life and Medical Sciences**]: Health, Medical information systems

General Terms

Algorithms

Keywords

Feature generalization, feature selection, sparse learning, augmented Lagrangian, spectral gradient descent, biomarkers

1. INTRODUCTION

Alzheimer's Disease (AD) is a severe neurodegenerative disorder that progresses over time. Electronic Health Records data such as Alzheimer's Disease Neuroimaging Initiative (ADNI) database provide valuable resources for conducting a longitudinal study of AD research. ADNI data are collected through regular hospital visits of AD patients after their first screening. In each visit, various measurements including cognitive scores (tests), lab tests, and brain images are collected for each patient, which serve as a large pool of potential biomarkers. Identification of important biomarkers that track the progression of AD is a central task towards a better understanding of the disease and the development of effective drugs. Many existing works build predictive models [30, 31, 36], perform longitudinal analysis [8, 39] and biomaker identification [42, 37, 11] directly over the raw features.

One of the fundamental challenges of biomarker identification is the gap between lower level features and higher level clinical concepts. Physicians and healthcare providers think and operate in terms of higher level clinical concepts, while the EHR data are heterogeneous sequences of features in a much lower level of granularities. The low level features are noisy (not all measurements are trustworthy), redundant (many features are highly correlated) and sparse (clinical events are known to be sparsely populated over time). Because of the above characteristics, the direct use of those low level features are problematic. One important ramification is the instability of feature selection against such noisy, redundant and sparse feature matrix. With a small perturbation of samples or feature values, the results of feature selection may vary significantly.

When it comes to the predictive modeling in longitudinal studies and healthcare analysis, the data sources are typi-

cally high dimensional. For example in the study of AD, popular biomarkers include brain images such as magnetic resonance imaging and positron emission tomography [27, 12], and genome information [14]. To deal with such high-dimensional data, sparse learning methods [32, 15] provide an effective tool that performs embedded feature selection via sparsity-inducing norms such as ℓ_1-norm [2]. Structural sparsity [9, 33, 16, 13] is recently introduced to control the structural patterns of the sparsity, exploring the inherent structures of the predictive modeling problems. The sparse learning has many successful applications in biomedical informatics and has produced new medical insights [7, 11, 33, 37, 39, 40, 42].

The ℓ_1-norm regularized methods such as Lasso [33] enjoy nice properties in terms of feature selection. However, theories on these methods often heavily lie on assumptions on the design matrix, i.e., the *irrepresentable condition* [38]. When using Lasso for feature selection in high dimensional problems, strongly correlated features usually result in poor model selection performance [4]. Unfortunately in many clinical studies and healthcare analysis problems this is usually the case; and thus the selected features are usually unstable under slight perturbations of the data. To deal with this instability problem, specific sparse learning methods are proposed to identify stable features via a large amount of boostrapping [22, 41], which is usually computationally expensive and cannot completely resolve the problem if the correlation is high. Recently, Bühlmann *et al.* proposed a two-stage approach that firstly learns feature groups by performing clustering on the design matrix and then performs Lasso on the new features constructed from the feature groups. Such two-stage approach has been shown to improve the condition of the design matrix used in the Lasso and is shown to have nice theoretical properties [3]. However, a separate feature group construction and feature selection may lead to suboptimal performance in terms of the stability of group selection and predictive performance. A more detailed discussion is given in Section 4.2.

Inspired by our experience on clinical predictive modeling and the aforementioned issues in the approach in [3], we propose an integrated approach, called *FeaFiner*, for feature construction and feature selection. The FeaFiner simultaneously generalizes lower level features into higher level clinical concepts and selects the predictive clinical concepts. Specifically, we propose a formulation for learning a sparse group structure matrix and a sparse prediction model via ℓ_1-regularization, and adopt an efficient block coordinate descent algorithm for solving the formulation. In many clinical research applications, non-overlapping groups are preferred for better interpretability. To this end, we further propose an improved formulation that learns non-overlapping feature groups via introducing additional orthogonality constraints to the formulation. However, the proposed formulation is challenging to solve due to the orthogonality constraints, non-convexity objective and non-smooth penalties. We solve our problem formulation using a novel augmented Lagrangian framework, recently developed in [19]. The key idea there is to solve this non-convex problem by a non-monotone spectral projected gradient method. The resulting approach is computationally efficient and also capable of producing solutions of high quality. We also present a generalization bound showing the consistency and the asymptotic behavior of the learning process of our proposed formulation.

We perform extensive experiments on both synthetic data and real datasets for the clinical studies of Alzheimer's disease. Results show that the proposed approach is capable of learning more stable feature groups than existing approaches while achieving superior predictive performance.

Notations: The element-wise ℓ_1-norm of a matrix $\mathbf{X}$ is denoted by $\|\mathbf{X}\|_1 = \sum_{i,j} |\mathbf{X}_{i,j}|$. We use $\mathbf{X} \geq 0$ to denote the elementwise non negativity ($\mathbf{X}_{i,j} \geq 0, \forall i, j$). $\mathbf{1}$ denotes the all-ones vector whose dimension is clear in context.

2. A FORMULATION FOR SIMULTANEOUS FEATURE GENERALIZATION AND SELECTION

In the healthcare analysis and clinical studies, one important task is to identify important risk factors and biomarkers that relate to a certain disease or health status of interest, and build predictive models from patient data. In the studies of Alzheimer's disease, for example, many researches focus on building predictive models that perform early detection and identify stable biomarkers that are related to the progression of the disease. Sparse learning is among the most popular techniques that are capable of simultaneously building parsimonious predictive models from training data and perform biomarker identification via embedded feature selection.

Consider a prediction task from n subjects with p features, where each feature is the value of a certain risk factor or the measurement of a biomarker. We denote the patients by data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ and their corresponding response by $\mathbf{y} \in \mathbb{R}^n$, where the response can be a continuous metric that indicates a certain clinical status of the patients. Given the training data $\mathbf{X}$ and $\mathbf{y}$ we aim to learn a predictive model. In this paper we consider only a linear model with a p-dimensional model vector denoted by $\mathbf{w} \in \mathbb{R}^p$ and the prediction given by $\hat{\mathbf{y}} = \mathbf{X}\mathbf{w} \in \mathbb{R}^n$. The classical sparse learning method Lasso [32] learns a sparse model by solving the following ℓ_1 regularized optimization problem:

$$\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2/n + \lambda_1 \|\mathbf{w}\|_1, \tag{1}$$

where λ_1 is a specified parameter that controls the sparsity of the model. By sparse model we mean there are many zeros in the model vector. If a feature has zero coefficient in the model, it is considered to be irrelevant to the prediction task and thus can be removed from the model. The ℓ_1-norm regularized formulations have been well studied over the last decade and widely used in many medical researches and clinical studies. Though the ℓ_1-norm based sparse learning methods yield high predictive power in practice, the learnt models are usually shown to be unstable if the training data is slightly perturbed [22, 3]. To tackle this problem, the authors of [3] proposed to firstly find correlated features via clustering and generate new higher-level features using the clustered groups. Sparse models are then built using the generated features. From the perspective of medical research and clinical studies, this approach is appealing because higher level features generalized from noisy and correlated raw features may be more stable and interpretable. It has been shown both theoretically and empirically that this approach gives more stable models. However, a separate feature generalization and selection may be suboptimal in terms of both predictive performance and quality of the obtained feature groups.

In this paper we propose a formulation that simultaneously performs feature generalization and selection to improve both the predictive performance and group quality. Before proceeding, we introduce some notations that will be used subsequently. For each group, we represent the group assignment information in a vector, and denote the ith group by $\mathbf{g}_i \in \mathbb{R}^p$. If the jth feature belongs to this group, then the jth component of $\mathbf{g}_i$ is non-zero and the relative magnitude represents the 'importance' of the feature in this group. The new feature generated from this group assignment is thus given by $\mathbf{X}\mathbf{g}_i$. Suppose we have k groups of features and we collectively denote the *group structure* by $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_k]$, and the generalized new features is then given by $\mathbf{XG}$. To make each group meaningful, we require the elements of $\mathbf{G}$ to be non-negative. We denote by $\mathbf{s} \in \mathbb{R}^k$ the new model vector associated with new feature groups. The resulting formulation of FeaFiner is given by:

$$\begin{aligned} \min_{\mathbf{s},\mathbf{G}} \ & \|\mathbf{XGs} - \mathbf{y}\|_2^2/n + \lambda_1\|\mathbf{s}\|_1 \\ \text{subject to: } & \mathbf{G} \geq 0, \|\mathbf{g}_i\|_1 \leq \theta, i = 1, \ldots, k, \end{aligned} \tag{2}$$

where θ is a specified parameter that controls the sparsity on the columns of $\mathbf{G}$, i.e., the number of features included in a group. In the solutions obtained by solving Eq. (2), the ℓ_1-norm lengths of most columns of $\mathbf{G}$ are exactly θ, providing a good interpretation for the group membership. The approach in [3] is a special case of our formulation in Eq. (2). Note that the elastic net also encourages group effects such that it tends to assign equal weights to the highly correlated features [43]. However, it cannot explicitly identify feature groups as does in (2).

The optimization problem (2) is generally non-convex since its objective function involves the product of two variables. A local optimal solution is thus often sought. One natural approach to solving problem (2) is by using the block coordinate descent algorithm, in which we alternatively solve $\mathbf{G}$ and $\mathbf{s}$ by fixing one variable and optimizing with respect to another. The details are as follows:

1) Given $\mathbf{G}$, we solve $\mathbf{s}$:

$$\mathbf{s}^+ = \arg\min_{\mathbf{s}} \ \|\mathbf{XGs} - \mathbf{y}\|_2^2/n + \lambda_1\|\mathbf{s}\|_1. \tag{3}$$

Solving $\mathbf{s}$ is a convex ℓ_1-regularized problem, which can be efficiently solved via the accelerated projected gradient method (APG) [24, 25].

2) Given $\mathbf{s}$, we solve $\mathbf{G}$:

$$\mathbf{G}^+ = \arg\min_{\mathbf{G}\in\mathcal{G}(\theta)} \ \|\mathbf{XGs} - \mathbf{y}\|_2^2/n, \tag{4}$$

where $\mathcal{G}(\theta) = \{\mathbf{G} : \mathbf{G} \geq 0, \|\mathbf{g}_i\|_1 \leq \theta, i = 1, \ldots, k\}$. Solving $\mathbf{G}$ is a constrained convex optimization problem, which again can also be solved via the APG method. The Euclidean projection onto the convex set $\mathcal{G}(\theta)$ can be efficiently solved with a linear time complexity [18]. The overall algorithm for solving formulation (2) is presented in Algorithm 1.

Algorithm 1 The block coordinate descent method for solving Eq. (2)

Input: $\mathbf{X}$, $\mathbf{y}$, Starting point $\mathbf{G}_0$.
Output: Grouping information $\mathbf{G}^*$, $\mathbf{s}^*$
  Initialize $\mathbf{G}^+ = \mathbf{G}_0$
  while not converge **do**
    Solve $\mathbf{s}^+ = \arg\min_{\mathbf{s}} \|\mathbf{XG}^+\mathbf{s} - \mathbf{y}\|_2^2/n + \lambda_1\|\mathbf{s}\|_1$.
    Solve $\mathbf{G}^+ = \arg\min_{\mathbf{G}\in\mathcal{G}(\theta)} \|\mathbf{XGs}^+ - \mathbf{y}\|_2^2/n$.
  end while
  Set $\mathbf{G}^* = \mathbf{G}^+$, $\mathbf{s}^* = \mathbf{s}^+$.

3. CONTROL OVERLAPPING IN GROUP LEARNING

The group structure obtained by solving formulation (2) may be largely overlapped because the proposed formulation does not impose any restriction on overlapping among the learnt groups. Nevertheless, in most clinical analysis applications practitioners often prefer less overlapped groups or even mutually exclusive groups, i.e., a particular biomarker should only belong to one feature group. To control the overlaps among groups, we impose the orthogonal constraints $\mathbf{g}_i^T\mathbf{g}_j = 0$ for all i, j in addition to the non-negative constraint $G \geq 0$. An immediate consequence of these constraints is that the resulting group assignments are mutually exclusive. For the simplicity of discussion, we normalize group assignments and assume that the columns of $\mathbf{G}$ are of length 1 with respect to ℓ_2 norm, which together with the orthogonality of the columns of $\mathbf{G}$ implies that $\mathbf{G}^T\mathbf{G} = \mathbf{I}$. In addition, we use the ℓ_1 norm regularization to control the sparsity on $\mathbf{G}$. Our improved formulation of non-overlapping FeaFiner is given by:

$$\begin{aligned} \min_{\mathbf{s},\mathbf{G}} \ & \|\mathbf{XGs} - \mathbf{y}\|_2^2/n + \lambda_S\|\mathbf{s}\|_1 + \lambda_G\|\mathbf{G}\|_1 \\ \text{subject to: } & \mathbf{G}^T\mathbf{G} = \mathbf{I}, \mathbf{G} \geq 0. \end{aligned} \tag{5}$$

3.1 Augmented Lagrangian Method

We observe that problem (5) is a constrained non-smooth optimization problem, which involves non-trivial constraint $\mathbf{G}^T\mathbf{G} = \mathbf{I}$. It is very natural to apply classical augmented Lagrangian method to solve (5). When applied to (5), augmented Lagrangian method needs to solve a sequence of subproblems in the form of

$$\min_{\mathbf{s},\mathbf{G}\geq 0} \mathcal{L}(\mathbf{s}, \mathbf{G}, \Lambda, \rho; \lambda_G, \lambda_s), \tag{6}$$

where $\mathcal{L}$ is the augmented Lagrangian function defined by

$$\begin{aligned} \mathcal{L}(\mathbf{G}, \mathbf{s}, \mathbf{\Lambda}, \rho; \lambda_G, \lambda_S) = & \|\mathbf{XGs} - \mathbf{y}\|_2^2/n + \lambda_S\|\mathbf{s}\|_1 \\ & + \lambda_G\|\mathbf{G}\|_1 - \langle \mathbf{\Lambda}, \mathbf{G}^T\mathbf{G} - \mathbf{I}\rangle + \tfrac{\rho}{2}\|\mathbf{G}^T\mathbf{G} - \mathbf{I}\|_F^2, \end{aligned}$$

$\mathbf{\Lambda} \in \mathbb{R}^{k\times k}$ is the Lagrange multiplier and $\rho \in \mathbb{R}^+$ is the penalty parameter, and $\|\cdot\|_F$ is the Frobenius norm. The augmented Lagrangian algorithm framework of solving Eq. (5) is given in Algorithm 2 (e.g., see [26]).

At the kth iteration, the main computational effort of Algorithm 2 lies in solving the augmented Lagrangian subproblem (6) with $\Lambda = \Lambda^{(k)}$ and $\rho = \rho^{(k)}$. This subproblem can be suitably solved by spectral projected gradient methods that were recently proposed in [35, 19] for solving a class of non-smooth optimization problems over a simple set. The discussion on one of these methods is postponed to Section 3.2.

As observed in our numerical experiment on Algorithm 2 for solving problem (5), the accumulation point of its generated sequence almost always violates some constraints of the problem, especially the orthogonal constraint $\mathbf{G}^T\mathbf{G} = \mathbf{I}$. The similar phenomenon has also been observed in [19] for solving a class of sparse PCA problems with orthogonality constraints. To overcome this drawback, the authors of [19] proposed a novel augmented Lagrangian method. And they showed that every accumulation point of the novel method

Algorithm 2 The classical augmented Lagrangian algorithm for solving orthogonal FeaFiner

Input: $\mathbf{X}$, $\mathbf{y}$, $\mathbf{G}^{(0)}$, $\mathbf{s}^{(0)}$, $\gamma > 1$, λ_G and λ_S.
Output: $\mathbf{G}^*$, $\mathbf{s}^*$.
Set $k = 0$, $\Lambda^{(0)} = \mathbf{1}\mathbf{1}^\mathbf{T}$, $\rho^{(0)} = 1$, $\mathcal{R}^{(0)} = (\mathbf{G}^{(0)})^T\mathbf{G}^{(0)} - \mathbf{I}$.
while not converge **do**
  Compute an approximate minimizer $(\mathbf{G}^{(k+1)}, \mathbf{s}^{(k+1)})$ for $\min_{\mathbf{s},\mathbf{G}\geq 0} \mathcal{L}(\mathbf{s}, \mathbf{G}, \Lambda^{(k)}, \rho^{(k)}; \lambda_G, \lambda_S)$.
  Compute residual $\mathcal{R}^{(k+1)} = (\mathbf{G}^{(k+1)})^T\mathbf{G}^{(k+1)} - \mathbf{I}$.
  if $\|\mathcal{R}^{(k+1)}\|_\infty < \eta\|\mathcal{R}^{(k)}\|_\infty$ **then**
    $\rho^{(k+1)} = \rho^{(k)}$, $\Lambda^{(k+1)} = \Lambda^{(k)} + \rho\mathcal{R}^{(k+1)}$;
  else
    $\rho^{(k+1)} = \gamma\rho^{(k)}$, $\Lambda^{(k+1)} = \Lambda^{(k)}$.
  end if
  Set $k := k + 1$.
end while
Set $\mathbf{G}^* = \mathbf{G}^k$ and $\mathbf{s}^* = \mathbf{s}^k$.

Algorithm 3 The novel augmented Lagrangian method for solving orthogonal FeaFiner

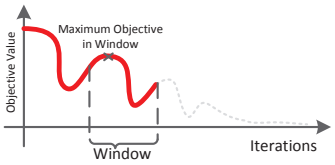
Input: $\mathbf{X}$, $\mathbf{y}$, $\gamma > 1$, $\sigma > 0$, λ_G and λ_S.
Output: $\mathbf{G}^*$, $\mathbf{s}^*$.
Compute an initial feasible solution $\mathbf{G}_0$ and $\mathbf{s}_0$ of problem (5). using the starting point strategy in Section 3.3.
Set $\Lambda^{(0)} = \mathbf{1}\mathbf{1}^\mathbf{T}$, $\rho^{(0)} = 1$
Compute $\tau = \mathcal{L}(\mathbf{s}_0, \mathbf{G}_0, \Lambda^{(0)}, \rho^{(0)}; \lambda_G, \lambda_S)$.
while not converge **do**
  Compute an approximate minimizer $(\mathbf{G}^{(k+1)}, \mathbf{s}^{(k+1)})$ for $\min_{\mathbf{s},\mathbf{G}\geq 0} \mathcal{L}(\mathbf{s}, \mathbf{G}, \Lambda^{(k)}, \rho^{(k)}; \lambda_G, \lambda_S)$
    such that $\mathcal{L}(\mathbf{s}, \mathbf{G}, \Lambda^{(k)}, \rho^{(k)}; \lambda_G, \lambda_S) \leq \tau$.
  Compute residual $\mathcal{R}^{(k+1)} = (\mathbf{G}^{(k+1)})^T\mathbf{G}^{(k+1)} - \mathbf{I}$.
  if $\|\mathcal{R}^{(k+1)}\|_\infty < \eta\|\mathcal{R}^{(k)}\|_\infty$ **then**
    $\rho^{(k+1)} = \rho^{(k)}$, $\Lambda^{(k+1)} = \Lambda^{(k)} + \rho\mathcal{R}^{(k+1)}$;
  else
    $\rho^{(k+1)} = \max(\gamma\rho^{(k)}, \|\Lambda^{(k)}\|_F^{1+\sigma})$, $\Lambda^{(k+1)} = \Lambda^{(k)}$.
  end if
  Set $k := k + 1$.
end while
Set $\mathbf{G}^* = \mathbf{G}^k$ and $\mathbf{s}^* = \mathbf{s}^k$.

must satisfy all constraints of the problem, and moreover under some suitable assumptions, each accumulation point is a KKT point of the problem (see Theorem 3.3 of [19]). It is not hard to verify that our problem (5) satisfies all the conditions required in Theorem 3.3 of [19]. Therefore, problem (5) can be suitably solved by the novel augmented Lagrangian method proposed in [19]. We present in Algorithm 3 the framework of this method for solving the orthogonal FeaFiner formulation (5). In contrast to Algorithm 2, Algorithm 3 has two novel features: one is that the Lagrangian function is bounded above by τ along the generated sequence; and another is that the penalty parameter $\rho^{(k)}$ grows faster than the magnitude of Lagrange multiplier $\Lambda^{(k)}$. In our experiment we observe that this novel method can perfectly recover the orthogonal group assignments on $\mathbf{G}$. Similar to Algorithm 2, the major computational part of Algorithm 3 lies in solving the augmented Lagrangian subproblem (6) with $\Lambda = \Lambda^{(k)}$ and $\rho = \rho^{(k)}$. We will discuss how to solve such subproblem efficiently in Section 3.2. Some other implementation details such as strategies for choosing good starting points, post-processing techniques, and pathwise solutions will be discussed in Section 3.3.

3.2 Solve Augmented Lagrangian Subproblem via Spectral Projected Gradient

As mentioned above, the major computational part of the novel augmented Lagrangian method for solving problem (5)

Figure 1: Illustration of non-monotone objective values in the spectral gradient descent. In the line-search we start from the maximum objective value of the previous $n_\mathcal{L}$ steps to find the next step size, where $n_\mathcal{L}$ is the window within which the objective values are not necessarily monotonically decreasing.



lies in solving the subproblem in the form of (6). We now discuss how to solve this subproblem efficiently. The two variables $\mathbf{G}, \mathbf{s}$ in (6) are coupled and thus bring non-convexity. Traditionally, block coordinate descent (BCD) method can be applied to solve this type of optimization problems [34]. Nevertheless, BCD may be easily trapped in a local minimizer in practice due to non-convexity and non-smoothness. Alternatively, we consider the $\mathbf{G}$ and $\mathbf{s}$ altogether and simultaneously solve these two variables, in the hope of better exploiting the internal structure of the optimization problem. Due to these considerations, we apply a non-monotone spectral projected gradient (SPG) method that was recently proposed in [19] for solving a class of non-smooth optimization problems over a simple set including problem (6) as a special case.

To apply the non-monotone SPG method to problem (6), we need the gradient of the smooth term in $\mathcal{L}$ which is denoted by $\tilde{\mathcal{L}}$, that is,

$$\tilde{\mathcal{L}}(\mathbf{G}, \mathbf{s}) = \frac{1}{n}\|\mathbf{X}\mathbf{G}\mathbf{s} - \mathbf{y}\|_2^2 - \langle \mathbf{\Lambda}, \mathbf{G}^T\mathbf{G} - \mathbf{I}\rangle + \frac{\rho}{2}\|\mathbf{G}^T\mathbf{G} - \mathbf{I}\|_F^2.$$

The gradient of $\tilde{\mathcal{L}}$ with respect to $\mathbf{G}$ is given by:

$$\begin{aligned}\nabla_\mathbf{G}\tilde{\mathcal{L}}(\mathbf{G}, \mathbf{s}) &= \nabla_\mathbf{G}\|\mathbf{X}\mathbf{G}\mathbf{s} - \mathbf{y}\|_2^2/n - \langle \mathbf{\Lambda}, \mathbf{G}^T\mathbf{G}\rangle + \frac{\rho}{2}\|\mathbf{G}^T\mathbf{G} - \mathbf{I}\|_F^2 \\ &= \frac{2}{n}\mathbf{X}^T\mathbf{X}\mathbf{G}\mathbf{s}\mathbf{s}^T - \frac{2}{n}\mathbf{X}^T\mathbf{y}\mathbf{s}^T - \mathbf{G}(\mathbf{\Lambda} + \mathbf{\Lambda}^T) \\ &\quad + 2\rho\mathbf{G}(\mathbf{G}^T\mathbf{G} - \mathbf{I}) \qquad (7)\end{aligned}$$

and its gradient with respect to $\mathbf{s}$ is:

$$\nabla_\mathbf{s}\tilde{\mathcal{L}}(\mathbf{G}, \mathbf{s}) = \nabla_\mathbf{s}\|\mathbf{X}\mathbf{G}\mathbf{s} - \mathbf{y}\|_2^2/n = \frac{2}{n}\mathbf{G}^T\mathbf{X}^T\mathbf{X}\mathbf{G}^T\mathbf{s} - \frac{2}{n}\mathbf{G}^T\mathbf{X}^T\mathbf{y}. \qquad (8)$$

During the line search procedure, we also need to solve the following proximal type ℓ_1-regularized problems:

$$\min_{\mathbf{X}\geq 0} \frac{1}{2}\|\mathbf{X} - \mathbf{G}\| + \nu\|\mathbf{X}\|_1, \quad \min_\mathbf{x} \frac{1}{2}\|\mathbf{x} - \mathbf{s}\| + \nu\|\mathbf{s}\|_1,$$

which have closed form solutions given by

$$\begin{aligned}\tilde{\mathcal{P}}_\nu(\mathbf{G}) &= \max(|\mathbf{G} - \nu|, 0), \qquad (9)\\ \mathcal{P}_\nu(\mathbf{s}) &= \text{sign}(\mathbf{s}) \cdot \max(|\mathbf{s} - \nu|, 0), \qquad (10)\end{aligned}$$

respectively, where $\cdot$ is the element-wise multiplication.

For the SPG method, one important issue is the choice of trial points and step size. Sicne problem (6) is non-convex, regular Armijo-type monotonic decreasing line search strategy may be too slow and also easily leads to a local solution. On the other hand, non-monotone line search strategy is more efficient and stable. Indeed, this strategy does not require the monotonic decrease of the objective value between consecutive steps, but rather requires a decrease within a

Algorithm 4 Spectral projected gradient method for solving augmented Lagrangian subproblem (6)

Input: $\mathbf{X}$, $\mathbf{y}$, ρ, Λ, $0 < \alpha_{\min} < \alpha_{\max}$, initial feasible point $(\mathbf{G}_0, \mathbf{s}_0)$, initial stepsize α_0, and an integer $n_{\mathcal{L}} > 0$.
Output: $\mathbf{G}_*$, $\mathbf{s}_*$.
Set $\mathbf{G}^- = \mathbf{G}_0$, $\mathbf{s}^- = \mathbf{s}_0$, $\alpha = \alpha_0$.
Compute $\nabla_{\mathbf{G}}\tilde{\mathcal{L}}(\mathbf{G}^-, \mathbf{s}^-)$, $\nabla_{\mathbf{s}}\tilde{\mathcal{L}}(\mathbf{G}^-, \mathbf{s}^-)$ using Eq. (7) and (8).
while $\varepsilon >$ tolerance **do**
Compute $\mathcal{L}_{\max}$ which is the maximum objective value over the latest $n_{\mathcal{L}}$ iterations.
Obtain $\mathbf{G}^+$, $\mathbf{s}^+$, $\nabla_{\mathbf{G}}\tilde{\mathcal{L}}(\mathbf{G}^+, \mathbf{s}^+)$, $\nabla_{\mathbf{s}}\tilde{\mathcal{L}}(\mathbf{G}^+, \mathbf{s}^+)$ using the non-monotone line search in Algorithm 5 with the initial stepsize α.
Compute $\Delta\mathbf{G} = \mathbf{G}^+ - \mathbf{G}^-$, $\Delta\mathbf{s} = \mathbf{s}^+ - \mathbf{s}^-$, $\Delta\mathbf{G}' = \nabla_{\mathbf{G}}\tilde{\mathcal{L}}(\mathbf{G}^+.\mathbf{s}^+) - \nabla_{\mathbf{G}}\tilde{\mathcal{L}}(\mathbf{G}^-, \mathbf{s}^-)$ and $\Delta\mathbf{s}' = \nabla_{\mathbf{s}}\tilde{\mathcal{L}}(\mathbf{G}^+, \mathbf{s}^+) - \nabla_{\mathbf{s}}\tilde{\mathcal{L}}(\mathbf{G}^-, \mathbf{s}^-)$.
Update initial stepsize $\alpha = \max(\alpha_{\min}, \min(\alpha_{\max}, \alpha'))$, where

$$\alpha' = \frac{\langle \Delta\mathbf{G}, \Delta\mathbf{G}\rangle + \langle \Delta\mathbf{s}, \Delta\mathbf{s}\rangle}{\langle \Delta\mathbf{G}, \Delta\mathbf{G}'\rangle + \langle \Delta\mathbf{s}, \Delta\mathbf{s}'\rangle}. \tag{11}$$

Compute $\varepsilon = \max(\|\mathcal{P}_{\lambda_G}(\mathbf{G}^+ - \nabla_{\mathbf{G}}\tilde{\mathcal{L}}(\mathbf{G}^+, \mathbf{S}^+)) - \mathbf{G}^+\|_\infty, \|\mathcal{P}_{\lambda_S}(\mathbf{s}^+ - \nabla_{\mathbf{s}}\tilde{\mathcal{L}}(\mathbf{G}^+, \mathbf{s}^+)) - \mathbf{s}^+\|_\infty)$.
Set $\mathbf{G}^- = \mathbf{G}^+$, $\mathbf{s}^- = \mathbf{s}^+$.
end while
Set $\mathbf{G}_* = \mathbf{G}^+$, $\mathbf{s}_* = \mathbf{s}^+$ and **return**.

Algorithm 5 Non-Monotone Armijo Line Search for Spectral Projected Gradient Method

Input: $\mathbf{G}^-, \mathbf{s}^-, \nabla_{\mathbf{G}}\tilde{\mathcal{L}}(\mathbf{G}^-, \mathbf{s}^-)$, $\nabla_{\mathbf{s}}\tilde{\mathcal{L}}(\mathbf{G}^-, \mathbf{s}^-), \mathcal{L}_{\max}$, $0 < \gamma < 1$, $0 < c < 1$, and initial stepsize α.
Output: $\mathbf{G}^+$, $\mathbf{s}^+$, $\nabla_{\mathbf{G}}\tilde{\mathcal{L}}(\mathbf{G}^+, \mathbf{s}^+)$, $\nabla_{\mathbf{s}}\tilde{\mathcal{L}}(\mathbf{G}^+, \mathbf{s}^+)$.
while true **do**
Solve $\mathbf{G}' = \tilde{\mathcal{P}}_{\alpha\lambda_g}(\mathbf{G}^- - \alpha\nabla_{\mathbf{G}}\tilde{\mathcal{L}}(\mathbf{G}^-, \mathbf{s}^-))$ via Eq. (9).
Solve $\mathbf{s}' = \mathcal{P}_{\alpha\lambda_g}(\mathbf{s}^- - \alpha\nabla_{\mathbf{s}}\tilde{\mathcal{L}}(\mathbf{G}^-, \mathbf{s}^-))$ via Eq. (10).
$\delta = c(\langle \mathbf{G}' - \mathbf{G}^-, \nabla_{\mathbf{G}}\tilde{\mathcal{L}}(\mathbf{G}^-, \mathbf{s}^-)\rangle + \langle \mathbf{s}' - \mathbf{s}^-, \nabla_{\mathbf{s}}\tilde{\mathcal{L}}(\mathbf{G}^-, \mathbf{s}^-)\rangle + \lambda_G\|\mathbf{G}'\|_1 + \lambda_s\|\mathbf{s}'\|_1 - \lambda_G\|\mathbf{G}^-\|_1 - \lambda_s\|\mathbf{s}^-\|_1)$.
if $\mathcal{L}(\mathbf{G}', \mathbf{s}', \mathbf{\Lambda}, \rho; \lambda_G, \lambda_S) \leq \mathcal{L}_{\max} + \delta$, **break**.
Update $\alpha = \alpha\gamma$.
end while
return $\mathbf{G}^+ = \mathbf{G}'$, $\mathbf{s}^+ = \mathbf{s}'$, $\nabla_{\mathbf{G}}\tilde{\mathcal{L}}(\mathbf{G}^+, \mathbf{s}^+)$, $\nabla_{\mathbf{s}}\tilde{\mathcal{L}}(\mathbf{G}^+, \mathbf{s}^+)$.

certain number of steps. The concept of non-monotone technique is illustrated in Figure 1. It was shown in [19] that under some suitable assumption the non-monotone SPG method has a linear convergence rate. The algorithm for solving Eq. (6) by non-monotone SPG method is presented in Algorithm 4, and the associated non-monotone line search subroutine is given in Algorithm 5.

The major difference between the spectral projected descent and the traditional projected gradient method lies in that: 1) Algorithm 2 requires the starting point to be feasible in order to converge according to the theoretical analysis in [19] (We will discuss the starting point strategies in Section 3.3.); 2) the initial step size is related to the approximate second-order information and given by the inverse Rayleigh quotient via Eq. (11).

3.3 Computational Issues

Starting Point. In [4] the clustering assignment is used to combine features. We expect to obtain a fairly good staring point of $\mathbf{G}$ from the assignment matrix. We normalize the assignment matrix such that the ℓ_2 norm length is 1 and denote it as $\mathbf{G}_{km}$. Then the starting value of $\mathbf{s}$ can be obtained by solving the least squares problem $\mathbf{s}_0 = \arg\min_{\mathbf{s}} \|\mathbf{X}\mathbf{G}_{km}\mathbf{s} - \mathbf{y}\|_2^2/n$, and from $\mathbf{s}_0$ we can obtain $\mathbf{G}_0$ by solving Eq. (4) with $\lambda_G = 0$.

Group Number. In many existing methods, the number of clusters or groups is obtained by either cross validation or domain knowledge. For FeaFiner, a meaningful starting point of $\mathbf{G}$ is the k-means clustering assignment matrix, thus we can select k by using heuristics for choosing the cluster number for k-means such as the simple rule $\sqrt{p/2}$ [20] or information criterion approaches such as AIC/BIC [5]. An alternative way of choosing the group number is by the expected group size, i.e., the number of features in each group, ignoring the sparsity. Given k groups, intuitively the expected group size is p/k. If fine-grained feature groups are needed, then a large k is needed and vice versa.

Post-Processing. Because of the orthogonality constraint on $\mathbf{G}$, the ℓ_1-norm sparsity on $\mathbf{G}$ may not behave as in unconstrained optimization problems. Normally we can view the ℓ_1-norm regularized problem as an equivalent constrained problem that requires the ℓ_1 length of columns of $\mathbf{G}$ to be less than or equal to a certain value. However, with the orthogonal constraint in Eq. (5), the ℓ_2-norm of columns of $\mathbf{G}$ is fixed to be 1, which indicates that the ℓ_1 length is implicitly lower-bounded. This means the solution $\mathbf{G}^*$ obtained by using Algorithm 2 may have some elements with very small values (e.g., less than $1e{-}5$) to ensure the unitary. Therefore we can add a post-processing step to set these small values to zeros and normalize the matrix after post-processing to be unitary, and then solve a Lasso problem in Eq. (3) to obtain the corresponding $\mathbf{s}$. Also, the post-processed solutions can again be used as the starting point in Algorithm 2 in the hope that a better local solution can be found.

Pathwise Solutions. The FeaFiner formulation in Eq. (5) has two sparse parameters λ_G and λ_S, which are typically estimated from data. In order to achieve high efficiency, we can obtain pathwise solutions via a *successive warm-start strategy*: for a fixed λ_S, we order a list of g parameter candidates for λ_G such that $\lambda_G^{(1)} < \ldots \lambda_G^{(g)}$ (from dense to sparse). We use $\{\mathbf{G}, \mathbf{s}\}_{\lambda_G}^{\lambda_S}$ to denote the solution obtained using λ_G and λ_S, and to compute $\{\mathbf{G}, \mathbf{s}\}_{\lambda_G(i+1)}^{\lambda_S}$ we use $\{\mathbf{G}, \mathbf{s}\}_{\lambda_G(i)}^{\lambda_S}$ as the starting point. We find that not only the pathwise solution strategy delivers higher computational efficiency, it also yields solutions that have higher quality than solving Eq. (5) independently for each parameter candidate. This strategy effectively prevents the algorithm from converging to inferior local solution. For convex sparse learning formulations a typical pathwise solution strategy requires a reversed order of parameter candidates (from sparse to dense), so that the solution space (from the constrained perspective) is very small at the very beginning to ensure efficiency. However, for the non-convex formulation of FeaFiner the pathwise strategy is reversed because in the dense case we know that the k-means solution $\mathbf{G}_{km}$ serves as a good starting point. In the experiments we use this pathwise strategy for parameter selection.

3.4 Theoretical Properties

In this section we provide some theoretical analysis of the proposed FeaFiner method. First we consider the following constrained reformulation of the FeaFiner with general Lipschiz continuous convex loss function ℓ (with Lipschiz constant L):

$$\begin{aligned} \min_{\mathbf{s},\mathbf{G}} \ & \frac{1}{n}\sum\nolimits_{i=1}^{n} \ell\left(\langle \mathbf{x}_i, \mathbf{G}\mathbf{s}\rangle, \mathbf{y}_i\right) \\ \text{s.t.} \ & \tau_{min} \leq \|\mathbf{G}\|_1 \leq \tau_{\max}^{\lambda_G}, \|\mathbf{s}\|_1 \leq \alpha, \mathbf{G}^T\mathbf{G} = \mathbf{I}, \mathbf{G} \geq 0, \end{aligned} \tag{12}$$

where $\tau_{min} = k \cdot \arg\min_{\|\mathbf{g}\|_2=1} \|\mathbf{g}\|_1$ and $\tau_{\max}^{\lambda_G}$ is related to the regularization parameter λ_G.

Due to the constraint $\mathbf{G}^T\mathbf{G} = \mathbf{I}$ and the element-wise non-negativity on $\mathbf{G}$, it immediately follows that Lasso is special case of FeaFiner when $k = n$, $\mathbf{G} = \mathbf{I}$ and $\mathbf{s}$ is solution of Lasso. We note that the original parameter space of the combined model $\mathbf{w} = \mathbf{G}\mathbf{s} \in \mathbb{R}^p$ is now expanded to a much larger parameter space $\mathbb{R}^{p,k} \times \mathbb{R}^{k,1}$, depending on K. The large parameter space may be a concern in practice because it is prone to overfitting. With $\mathbf{G}^T\mathbf{G} = \mathbf{I}$, the ℓ_1 constraint on $\mathbf{s}$ provides effective regularization on the resulting model $\mathbf{w} = \mathbf{G}\mathbf{s}$:

$$\|\mathbf{w}\|_F^2 = \|\mathbf{G}\mathbf{s}\|_F^2 = \mathrm{tr}\left(\mathbf{s}^T\mathbf{G}^T\mathbf{G}\mathbf{s}\right) = \mathrm{tr}\left(\mathbf{s}^T\mathbf{s}\right)$$
$$= \|\mathbf{s}\|_2^2 \le \|\mathbf{s}\|_1^2 \le \alpha^2.$$

We next show that the generalization error of the optimizer to the problem in Eq. (12) can be bounded and is related to the condition of the design matrix $\mathbf{X}$. Let $\mathcal{G}_k = \{\mathbf{G} \in \mathbb{R}^{p\times k} : \tau_{min} \le \|\mathbf{G}\|_1 \le \tau_{\max}^{\lambda_G}, \mathbf{G}^T\mathbf{G} = \mathbf{I}, \mathbf{G} \ge 0\}$ and $\mathcal{S} = \{\mathbf{s} \in \mathbb{R}^k : \|\mathbf{s}\|_1 \le \alpha\}$. Given any $\mathbf{G}$ and $\mathbf{s}$, we denote the expected risk as:

$$\mathbb{E}(\mathbf{G}, \mathbf{s}) = \mathbb{E}_{(\boldsymbol{x},\mathbf{y})\sim\boldsymbol{\mu}}[\ell(\langle \mathbf{G}\mathbf{s}, \boldsymbol{x}\rangle, \mathbf{y})].$$

Let $\mathbf{G}^*$ and $\mathbf{s}^*$ be the optimal solution that minimizes the expected risk:

$$(\mathbf{G}^*, \mathbf{s}^*) = \operatorname*{arg\,min}_{\mathbf{G}\in\mathcal{G}_K, \mathbf{s}\in\mathcal{S}} \mathbb{E}(\mathbf{G}, \mathbf{s}) = \operatorname*{arg\,min}_{\mathbf{G}\in\mathcal{G}, \mathbf{s}\in\mathcal{S}} \mathbb{E}_{(x,y)\sim\mu_t}[\ell(\langle \mathbf{G}\mathbf{s}, x\rangle, y)],$$

Also, given data $\mathbf{Z} = (\mathbf{X}, \mathbf{y})$, the empirical risk is defined as:

$$\hat{\mathbb{E}}(\mathbf{G}, \mathbf{s}|\mathbf{Z}) = \frac{1}{n}\sum_{i=1}^{n}[\ell(\langle \mathbf{G}\mathbf{s}, x_i\rangle, y_i)].$$

and let $\mathbf{G}^*_{(\mathbf{Z})}$ and $\mathbf{s}^*_{(\mathbf{Z})}$ be the optimal solution that minimizes the empirical risk: $\left(\mathbf{G}^*_{(\mathbf{Z})}, \mathbf{s}^*_{(\mathbf{Z})}\right) = \arg\min_{\mathbf{G}\in\mathcal{G}, \mathbf{s}\in\mathcal{S}} \hat{\mathbb{E}}(\mathbf{G}, \mathbf{s}|\mathbf{Z})$. The asymptotic convergence of the learning process is given in the following theorem:

THEOREM 3.1. *Let $\delta > 0$ and let μ be probability measure on $\mathbb{R}^d \times \mathbb{R}$. With probability of at least $1 - \delta$ in the draw of $\mathbf{Z} \sim \mu^n$, we have:*

$$\mathbb{E}(\mathbf{G}^*_{(\mathbf{Z})}, \mathbf{s}^*_{(\mathbf{Z})}) - \mathbb{E}(\mathbf{G}^*, \mathbf{s}^*) \le 2L\alpha\sqrt{\frac{2\mathcal{C}_1(\mathbf{X})(k+12)}{n}}$$
$$+ 2L\alpha\sqrt{\frac{8\mathcal{C}_\infty(\mathbf{X})\ln(2k)}{n}} + \sqrt{\frac{8\ln 4/\delta}{n}},$$

where $\mathcal{C}_1(\mathbf{X}) = \|\hat{\Sigma}(\mathbf{X})\|_* := \mathrm{tr}\left(\hat{\Sigma}(\mathbf{X})\right) = \sum_{i=1}^d \lambda_i(\hat{\Sigma}(\mathbf{X}))$ is the trace of the empirical covariance matrix, $\mathcal{C}_\infty(\mathbf{X}) = \|\hat{\Sigma}(\mathbf{X})\|_\infty := \lambda_{\max}\left(\hat{\Sigma}(\mathbf{X})\right)$ where $\lambda_{\max}$ is the largest eigenvalue, and $\hat{\Sigma}(\mathbf{X})$ is the empirical covariance matrix, i.e., $\hat{\Sigma}(\mathbf{X}) = \frac{1}{n}\mathbf{X}^T\mathbf{X} \in \mathbb{R}^{d\times d}$.

The proof structure is similar to that of [21] and to make the paper self-contained we include the detailed proof in the supplemental materials [1]. This theorem provides important insight into the proposed formulation in Eq. (5): 1) when $n \to \infty$, we have $\mathbb{E}(\mathbf{G}^*_{(\mathbf{Z})}, \mathbf{s}^*_{(\mathbf{Z})}) - \mathbb{E}(\mathbf{G}^*, \mathbf{s}^*)$ converges asymptotically to 0. 2) the convergence is related to the condition of design matrix $\mathbf{X}$ via $\mathcal{C}_\infty(\mathbf{X}), \mathcal{C}_1(\mathbf{X})$. If the design matrix has a low-rank structure, which gives a small $\mathcal{C}_1(\mathbf{X})$, then it achieves fast convergence.

4. EMPIRICAL STUDIES

4.1 Synthetic Studies

In the synthetic experiment we study the efficiency of the proposed non-orthogonal FeaFiner (N-FeaFiner) in Algorithm 1 and orthogonal FeaFiner (O-FeaFiner) in Algorithm 2, and evaluate the quality of the group structures obtained by the two algorithms.

Data Generation. We generate the data in the following way. Given the problem size n, p, we firstly generate a block diagonal covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{k\times k}$, with the block size $\lfloor p/k \rfloor$. Within each block, we set the diagonal elements to be 1 and off-diagonal to be 0.9. The design matrix $\mathbf{X}$ is then sampled from $\mathcal{N}(0, \mathbf{\Sigma})$. The matrix $\mathbf{G} \in \mathbb{R}^{p\times k}$ is generated according to the group structure defined by the covariance matrix. Suppose the p features are partitioned into k groups $\{I_1, I_2, \ldots, I_k\}$. The ith column of matrix $\mathbf{G}$ is sampled as follows: $\mathbf{G}_{i,j} \sim U(0,1)$ if $j \in I_i$ and $\mathbf{G}_{i,j} = 0$ otherwise. $\mathbf{s}$ is sampled from $N(0,1)$ with half of the entries set to 0. Finally, we construct the response vector $\mathbf{y} = \mathbf{X}\mathbf{G}\mathbf{s} + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 10^{-3})$.

Computational Efficiency. We first compare the computational cost of N-FeaFiner and O-FeaFiner. We set the precision of the outer-iteration of both algorithms to be 10^{-3} and the precision of the inner-iteration to be 10^{-6}. For N-FeaFiner, the inner-iteration is the ℓ_1-regularized/constrained solvers of $\mathbf{s}$ and $\mathbf{G}$, and for O-FeaFiner, the inner-iteration solves the SPG. We control the sparsity parameters so that the solutions of the two algorithms have approximately the same density (density is given by the number of non-zero elements divided by the number of total elements, and the density for $\mathbf{G}$ and $\mathbf{s}$ are 0.1 and 0.5 respectively). We perform experiments in three settings: 1) fix $n = 100, p = 500$ and vary $k = 10 : 10 : 100$; 2)fix $n = 100, k = 20$ and $d = 100 : 100 : 1000$; 3) fix $d = 300, k = 10$, and $n = 100 : 100 : 1000$. We repeat these experiments for 100 times and report the average time in Figure 2. We observe that 1) in general O-FeaFiner has advantage over N-FeaFiner in terms of computational cost; 2) the costs of both methods are linear w.r.t. the sample size n; 3) when increasing the dimensionality of the original feature space p, the costs of both methods increase sublinearly; 4) when increasing the group size k, the time cost of N-FeaFiner increases linearly while for O-FeaFiner the cost increases sublinearly w.r.t. the group number.

Group Overlap. The major difference between N-FeaFiner and O-FeaFiner is the non-overlapping constraint on the group structure $\mathbf{G}$ in O-FeaFiner. We perform experiments to study the group structures obtained by the two methods.

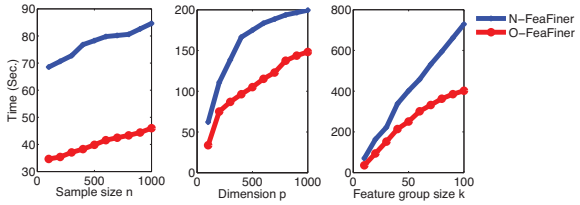


Figure 2: Comparison of computational complexity between the non-orthogonal FeaFiner (N-FeaFiner) and the improved orthogonal FeaFiner (O-FeaFiner) with varying sample size (left) n, dimensionality p (middle) and group number k (right).

Table 1: The demographic information of the ADNI dataset used in this study.

MMSE	Samples(Train/Test)	Validation Set
M06	434 (290/44)	214
M12	430 (387/43)	212
M24	381 (343/38)	188
M36	261 (235/26)	128
ADAS-Cog	Samples(Train/Test)	Validation Set
M06	434 (391/43)	214
M12	427 (385/42)	211
M24	378 (341/37)	186
M36	253 (228/25)	124

Similar to the previous experiment, we construct a toy data of size $n = 50, p = 12, k = 3$. We train predictive models using the two methods and present the learned group structures $\mathbf{G}$ in Figure 3. We observe that the O-FeaFiner algorithm can perfectly recover the location of the non-zero elements, while the group structure obtained by N-FeaFiner introduces irrelevant assignments and the groups overlap.

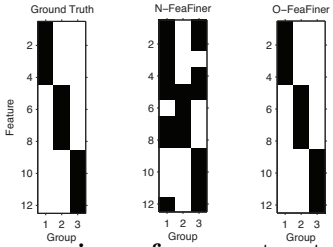


Figure 3: Comparison of group structures obtained by N-FeaFiner (middle) and O-FeaFiner (right) on the toy data. The O-FeaFiner can perfectly recover the ground truth (left).

4.2 Identifying Biomarkers for AD

In this experiment we apply O-FeaFiner to analyze the feature groups and build effective predictive models on the ADNI dataset[1]. In the ADNI project, images such as magnetic resonance imaging (MRI) scans and important cognition-related clinical measurements such as Mini Mental State Examination (MMSE) scores and Alzheimer's Disease Assessment Scale-cognitive subscores (ADAS-Cog) are obtained from selected patients repeatedly over a 6-month or 1-year interval. The MMSE scores and ADAS-Cog scores are shown to be correlated with the underlying AD pathology and progressive deterioration of functional ability [28].

Experimental Settings. In this study we perform experiments to build predictive models from the 306 low-level features to predict the MMSE and ADAS-Cog scores at future time points (M06, M12, M24, M36). The prediction of one type of cognitive score at one time point is a regression task, and therefore there are in total 8 regression tasks. The low-level features are extracted from the baseline MRI brain scans of the patients, and a detailed list of the features is given in the supplemental materials [1]. The prediction at a particular time point makes use of all the samples that have the MMSE score at the particular time point as well as the baseline MRI scans. We split the samples into two parts: one third of the samples are served as an independent validation dataset used to estimate the tunable parameters and 90% of the remaining samples are used to build the model and 10% of the remaining samples are used to test the predictive models. The detailed demographic information of the data is given in Table 1. In the experiment we normalize the two scores for all the samples such that after the normalization their values are in the range of $[0, 1]$. We randomly split the training and testing data, and repeat the experiment for 10 times. We compare the proposed orthogonal FeaFiner with two baseline methods:

- Lasso (SLEP implementation [17]). Lasso in nature is a much easier problem than FeaFiner in the sense that it only pursuits high predictive performance, and does not identify feature groups. We show that FeaFiner achieves equal or even higher performance than Lasso.
- Cluster Representative Lasso (CRL) [3] performs clustering first then uses lasso to select from the resulting clusters. We show that the performance of the FeaFiner is significantly better than CRL, because FeaFiner can jointly perform feature grouping and selection instead of using a two-stage approach like CRL.

To study the effects of varying group number k, we manually choose three values (12 which is the simple rule of thumb number [20], 30 and 50) of k in CRL and FeaFiner methods and report results independently. We evaluate the three methods in terms of their predictive performance, model stability (for CRL and FeaFiner the models are built using grouped features) and the stability of the learned groups (for CRL and FeaFiner only).

Predictive Performance. We evaluate the performance of the algorithms by the coefficient of determination (R^2) [29], which is widely used in the regression analysis of medical studies. Given the ground truth target vector y and its corresponding prediction $\hat{y}$, the R^2 metric is defined by: $R^2 = 1 - \left(\|y - \hat{y}\|_2^2 / \|y - \bar{y}\|_2^2\right)$, where $\bar{y}$ is a vector whose elements are the mean of y. In Table 2 we report the average experimental results on MMSE and ADAS-Cog prediction in terms of predictive performance. We find that Lasso and the proposed FeaFiner method achieve high predictive performance, while the CRL method does not perform well in most cases.

Table 2: Comparison of predictive performance of the proposed approach (O-FeaFiner) and existing approaches (Lasso and CRL) on MMSE and ADAS-Cog prediction in terms of coefficient of determination (R^2). Higher R^2 indicates better predictive performance.

MMSE	M06	M12	M24	M36
Lasso	0.5479	0.4773	0.5892	0.4106
CRL (12)	0.2155	0.2395	0.2743	0.1332
CRL (30)	0.2819	0.2816	0.3216	0.2081
CRL (50)	0.2856	0.2559	0.3879	0.3223
FeaFiner (12)	0.4619	0.4798	**0.5998**	0.3724
FeaFiner (30)	0.5562	**0.4818**	0.5896	0.3731
FeaFiner (50)	**0.5628**	0.4579	0.5561	**0.4203**
ADAS-Cog	M06	M12	M24	M36
Lasso	0.4969	**0.5581**	0.5170	**0.4438**
CRL (12)	0.2695	0.2950	0.3232	0.2167
CRL (30)	0.2860	0.3183	0.4488	0.3204
CRL (50)	0.3612	0.4374	0.4533	0.1563
FeaFiner (12)	**0.5282**	0.5385	**0.5484**	0.3275
FeaFiner (30)	0.5036	0.5303	0.5342	0.4355
FeaFiner (50)	0.5106	0.5447	0.5321	0.3391

Model Stability. To evaluate the stability of models, we define the following metric: for each feature f_i we use the inclusion function $\mathcal{I}(f_i)$ to indicate if this feature is included in the model ($\mathcal{I}(f_i) = 1$ if the feature is included and $\mathcal{I}(f_i) = 0$ otherwise) and $\text{var}(\mathcal{I}(f_i))$ to denote the variance of the inclusion w.r.t. models obtained from random splittings, and the *feature variance* is defined by $\sum_{i=1}^{p} \text{var}(\mathcal{I}(f_i))/p$. If a

[1]Available at http://adni.loni.ucla.edu/

Table 3: Comparison of model stability of the proposed approach (O-FeaFiner) and existing approaches (Lasso and CRL) on MMSE and ADAS-Cog prediction in terms of feature variance. A lower feature variance indicates that the models are more stable.

MMSE	M06	M12	M24	M36
Lasso	0.1021	0.1464	0.2565	0.1963
CRL (12)	0.1303	0.1733	0.1652	0.2852
CRL (30)	0.1700	0.1631	0.1570	0.1615
CRL (50)	0.1597	0.1234	0.1361	0.1080
FeaFiner (12)	0.1975	0.1451	0.0864	0.1670
FeaFiner (30)	0.1469	0.0604	**0.0532**	0.0742
FeaFiner (50)	**0.0378**	**0.0330**	0.0553	**0.0616**
ADAS-Cog	M06	M12	M24	M36
Lasso	0.1832	0.1313	0.2066	0.2722
CRL (12)	0.2155	0.2071	0.2836	0.2481
CRL (30)	0.1362	0.1565	0.1377	0.1426
CRL (50)	0.1038	0.1026	0.1214	0.1209
FeaFiner (12)	0.1329	0.0822	0.0762	0.1565
FeaFiner (30)	**0.0335**	**0.0401**	0.0767	**0.0363**
FeaFiner (50)	0.0544	0.0575	**0.0567**	0.0459

Table 4: Comparison of group stability of the proposed approach (O-FeaFiner) and the existing approach (CRL) on MMSE and ADAS-Cog prediction in terms of group variance). A lower group variance indicates that the groups used to generalize features are more stable.

MMSE	M06	M12	M24	M36
CRL (12)	0.1369	0.1308	0.1245	0.1376
CRL (30)	0.0525	0.0494	0.0532	0.0596
CRL (50)	0.0333	0.0331	0.0322	0.0352
FeaFiner (12)	0.1506	0.0077	**0.0077**	0.0178
FeaFiner (30)	0.0166	**0.0069**	0.0540	0.0213
FeaFiner (50)	**0.0109**	0.0070	0.0338	**0.0092**
ADAS-Cog	M06	M12	M24	M36
CRL (12)	0.1448	0.1342	0.1303	0.1267
CRL (30)	0.0548	0.0550	0.0571	0.0534
CRL (50)	0.0326	0.0331	**0.0320**	0.0340
FeaFiner (12)	0.0078	**0.0225**	0.1424	0.0437
FeaFiner (30)	0.0619	0.0600	0.0584	**0.0157**
FeaFiner (50)	**0.0061**	0.0355	0.0342	0.0198

feature is included or excluded by all models of random splittings, the feature variance is 0. For CRL and FeaFiner we report the variance of the features generated after grouping. The group assignments and models across different random splittings are aligned using the best correlation. In Table 3 we report the model stability of all competing methods on MMSE and ADAS-Cog predictions. We find that the models built by Lasso are not stable while CRL and FeaFiner produce much more stable models when $k = 30$ and $k = 50$ (especially at time points M24 and M36). However, using an improper k may yield unstable models for both methods.

Group Stability. To evaluate the stability of the learned groups, we define the following metric: denote the group assignment vector for group i obtained from the qth random splitting experiment by $\mathbf{g}_i^{(q)}$, and the *group variance* is defined by:

$$\sum_{i=1}^{k} \left(\sum_{q \neq r} \mathbf{I}(\mathcal{I}(\mathbf{g}_i^{(q)}) \neq \mathcal{I}(\mathbf{g}_i^{(r)})) / \sum_{q \neq r} 1 \right) / k,$$

where $\mathbf{I}$ is the standard indicator function. The group variance measures how likely the group assignment of one variable changes over different random splittings. We report the average group stability on MMSE and ADAS-Cog prediction in Table 4. We see that the groups learned by FeaFiner are in general more stable than CRL.

Table 5: Examples of high-level feature groups obtained by the proposed FeaFiner algorithm ($k = 50$)

Feature Group	Raw Feature Name	Group Weight
MMSE Group 1	Vol.(CO) L.TemporalPole	0.1847
	Vol.(CO) R.TemporalPole	0.1679
	Suf. Area L.TemporalPole	0.1595
	Suf. Area R.TemporalPole	0.1624
	Suf. Area L.Entorhinal	0.1682
	Suf. Area R.Entorhinal	0.1573
MMSE Group 2	Vol.(WM) L.Hippocampus	0.2513
	Vol.(WM) R.Hippocampus	0.2545
	Vol.(WM) L.Amygdala	0.2451
	Vol.(WM) R.Amygdala	0.2491
MMSE Group 3	CT Avg. L.Sup.Frontal	0.4912
	CT Avg. R.Pos.Cingulate	0.2857
	CT Avg. R.IsthmusCingulate	0.2231
ADAS Group 1	Baseline MMSE	0.7820
	Suf. Area R.Fusiform	0.0830
	Vol.(CO) L.Ros.Mid.Frontal	0.0307
	Vol.(CO) R.Ros.Mid.Frontal	0.1043
ADAS Group 2	CT Avg. L.Cuneus	0.3490
	CT Std. L.Cuneus	0.3217
	Vol.(CO) L.Cuneus	0.3294
ADAS Group 3	CT Avg. L.Tra.Temporal	0.2490
	CT Avg. R.Tra.Temporal	0.2505
	CT Std. L.Tra.Temporal	0.2506
	CT Std. R.Tra.Temporal	0.2499

Feature Group Analysis. We list some examples of high-level feature groups learned by FeaFiner ($k = 50$) in Table 5. A detailed list of feature groups from different tasks are available in the supplemental materials [1]. We observe several interesting patterns in the learned groups. First of all we find that many feature groups exhibit bilaterally symmetric patterns. For a certain brain area there are two low-level features, i.e., one for the left hemisphere and one for the right. If the feature from one hemisphere is included in a group, then the corresponding counterpart on the other hemisphere is also likely to be included in the same feature group. This agrees with the observations from many medical researches, in which reductions on many bilaterally symmetric brain regions were found in the AD patients [10, 23]. Note that in some groups we also find interesting asymmetric groups such as MMSE group 3 and ADAS group 2 in Table 5. The asymmetric feature such as Cingulate has been identified in some recent studies on asymmetry disease biomarkers [6]. We have several low level features for a particular brain area (e.g., volumes, surface area and cortical thickness average/standard deviation). We find from our experiments that features from the same brain area are likely to belong to the same group (e.g., ADAS Group 3). We also notice that in many feature groups the weights of the low-level features are not equally distributed; for example in MMSE Groups 1, 3 and ADAS Group 1. This indicates that in the same feature group some features may contribute more to the prediction than the others, and existing clustering-based methods such as CRL are not able to obtain groups of such kind.

5. CONCLUSION

In this paper we propose an integrated approach called *FeaFiner* for feature construction by simultaneously identifying a feature grouping structure which projects data from a high dimensional feature space to a low-dimensional and interpretable feature space, and learning a sparse model on the low-dimensional space. We propose two formulations: N-FeaFiner for learning overlapped groups and O-FeaFiner for learning mutually exclusive groups. We propose novel al-

gorithms for solving the two problems. We have performed extensive experiments on both synthetic and real datasets to evaluate the proposed algorithms, and results show that the proposed method learns clinically meaningful feature groups, and demonstrates promising predictive performance on real medical data sets. One of our future works is to apply the proposed algorithm to other biomedical applications.

Acknowledgement

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

6. REFERENCES

[1] `www.public.asu.edu/~jye02/FeaFiner`.

[2] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Convex optimization with sparsity-inducing norms. *Opt. for Mach. Learn.*, pages 19–53, 2011.

[3] P. Bühlmann, P. Rütimann, S. van de Geer, and C. Zhang. Correlated variables in regression: clustering and sparse estimation. *arXiv preprint arXiv:1209.5908*, 2012.

[4] P. Bühlmann and S. Van De Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer, 2011.

[5] K. P. Burnham and D. R. Anderson. Multimodel inference understanding aic and bic in model selection. *Soc. Met. & Res.*, 33(2):261–304, 2004.

[6] R. Cabeza. Hemispheric asymmetry reduction in older adults: the harold model. *Psych. and ag.*, 17(1):85, 2002.

[7] A. Dobra, C. Hans, B. Jones, J. R. Nevins, G. Yao, and M. West. Sparse graphical models for exploring gene expression data. *J. of Multi. Ana.*, 90(1):196–212, 2004.

[8] S. Duchesne, A. Caroli, C. Geroldi, D. L. Collins, and G. B. Frisoni. Relating one-year cognitive change in mild cognitive impairment to baseline mri features. *Neuroimage*, 47(4):1363–1370, 2009.

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

[10] B. Horwitz, C. L. Grady, N. Schlageter, R. Duara, and S. Rapoport. Intercorrelations of regional cerebral glucose metabolic rates in alzheimer's disease. *Brain research*, 407(2):294–306, 1987.

[11] S. Huang, J. Li, L. Sun, J. Liu, T. Wu, K. Chen, A. Fleisher, E. Reiman, and J. Ye. Learning brain connectivity of alzheimers disease from neuroimaging data. *NIPS*, 22:808–816, 2009.

[12] C. R. Jack, M. A. Bernstein, N. C. Fox, P. Thompson, G. Alexander, D. Harvey, B. Borowski, P. J. Britson, J. L Whitwell, C. Ward, et al. The alzheimer's disease neuroimaging initiative (adni): Mri methods. *J. of Mag. Res. Imag.*, 27(4):685–691, 2008.

[13] L. Jacob, G. Obozinski, and J. Vert. Group lasso with overlap and graph lasso. In *ICML*, pages 433–440, 2009.

[14] M. Kamboh, F. Demirci, X. Wang, R. Minster, M. Carrasquillo, V. Pankratz, S. Younkin, A. Saykin, G. Jun, C. Baldwin, et al. Genome-wide association study of alzheimer's disease. *Trans. Psyc.*, 2(5):e117, 2012.

[15] J. Liu, J. Chen, and J. Ye. Large-scale sparse logistic regression. In *KDD*, pages 547–556, 2009.

[16] J. Liu, S. Ji, and J. Ye. Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization. In *UAI*, pages 339–348, 2009.

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

[18] J. Liu and J. Ye. Efficient euclidean projections in linear time. In *ICML*, pages 657–664, 2009.

[19] Z. Lu and Y. Zhang. An augmented lagrangian approach for sparse principal component analysis. *Math. Prog.*, pages 1–45, 2011.

[20] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Academic Press, 1979.

[21] A. Maurer, M. Pontil, and B. Romera-Paredes. Sparse coding for multitask and transfer learning. In *ICML*, pages 343–351, 2013.

[22] N. Meinshausen and P. Bühlmann. Stability selection. *J. of the Roy. Stat. Soc.: Series B (Stat. Meth.)*, 72(4):417–473, 2010.

[23] J. Moossy, G. S. Zubenko, A. J. Martinez, and G. R. Rao. Bilateral symmetry of morphologic lesions in alzheimer's disease. *Arch. of Neuro.*, 45(3):251, 1988.

[24] A. Nemirovsky and D. Yudin. Problem complexity and method efficiency in optimization. 1983.

[25] Y. Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.

[26] J. Nocedal and S. Wright. *Numerical optimization*. Springer verlag, 1999.

[27] A. Nordberg et al. Pet imaging of amyloid in alzheimer's disease. *Lancet neurology*, 3(9):519, 2004.

[28] J. R. Petrella, R. E. Coleman, and P. M. Doraiswamy. Neuroimaging and early diagnosis of alzheimer disease: A look to the future1. *Radiology*, 226(2):315–336, 2003.

[29] R. G. Steel, J. H. Torrie, and D. A. Dickey. Principles and procedures of statistics. *Principles and procedures of statistics*, 1960.

[30] C. M. Stonnington, C. Chu, S. Klöppel, C. R. Jack Jr, J. Ashburner, R. S. Frackowiak, et al. Predicting clinical scores from magnetic resonance scans in alzheimer's disease. *Neuroimage*, 51(4):1405, 2010.

[31] P. M. Thompson, K. M. Hayashi, G. I. de Zubicaray, A. L. Janke, S. E. Rose, J. Semple, M. S. Hong, D. H. Herman, D. Gravano, D. M. Doddrell, et al. Mapping hippocampal and ventricular change in alzheimer disease. *Neuroimage*, 22(4):1754–1766, 2004.

[32] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. of the Roy. Stat. Soc. Series B (Meth.)*, pages 267–288, 1996.

[33] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *J. of the Roy. Stat. Soc.: Series B (Stat. Meth.)*, 67(1):91–108, 2004.

[34] P. Tseng and S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Math. Prog.*, 117(1):387–423, 2009.

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

[36] L. Yuan, Y. Wang, P. M. Thompson, V. A. Narayan, and J. Ye. Multi-source feature learning for joint analysis of incomplete multiple heterogeneous neuroimaging data. *NeuroImage*, 61(3):622–632, 2012.

[37] D. Zhang and D. Shen. Predicting future clinical changes of mci patients using longitudinal and multimodal biomarkers. *PloS one*, 7(3):e33182, 2012.

[38] P. Zhao and B. Yu. On model selection consistency of lasso. *JMLR*, 7(2):2541, 2007.

[39] J. Zhou, J. Liu, V. A. Narayan, and J. Ye. Modeling disease progression via fused sparse group lasso. In *KDD*, pages 1095–1103, 2012.

[40] J. Zhou, J. Liu, V. A. Narayan, and J. Ye. Modeling disease progression via multi-task learning. *NeuroImage*, 78:233–248, 2013.

[41] J. Zhou, J. Sun, Y. Liu, J. Hu, and J. Ye. Patient risk prediction model via top-k stability selection. In *SDM*, pages 55–63, 2013.

[42] J. Zhou, L. Yuan, J. Liu, and J. Ye. A multi-task learning formulation for predicting disease progression. In *KDD*, pages 814–822, 2011.

[43] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *J. of the Roy. Stat. Soc. Series B (Meth.)*, 67:301–320, 2005.