Multi-Source Learning with Block-wise Missing Data for Alzheimer's Disease Prediction

Shuo Xiang[1,2], Lei Yuan[1,2], Wei Fan[3], Yalin Wang[1], Paul M. Thompson[4], Jieping Ye[1,2]
[1]Computer Science and Engineering, ASU, Tempe AZ 85287
[2]Center for Evolutionary Medicine and Informatics, The Biodesign Institute, ASU, Tempe AZ 85287
[3]Huawei Noah's Ark Lab, Hong Kong, China
[4]Laboratory of Neuro Imaging, Department of Neurology, UCLA, Los Angeles, CA 90095

ABSTRACT

With the advances and increasing sophistication in data collection techniques, we are facing with large amounts of data collected from multiple heterogeneous sources in many applications. For example, in the study of Alzheimer's Disease (AD), different types of measurements such as neuroimages, gene/protein expression data, genetic data etc. are often collected and analyzed together for improved predictive power. It is believed that a joint learning of multiple data sources is beneficial as different data sources may contain complementary information, and feature-pruning and data source selection are critical for learning interpretable models from high-dimensional data. Very often the collected data comes with block-wise missing entries; for example, a patient without the MRI scan will have no information in the MRI data block, making his/her overall record incomplete. There has been a growing interest in the data mining community on expanding traditional techniques for single-source complete data analysis to the study of multi-source incomplete data. The key challenge is how to effectively integrate information from multiple heterogeneous sources in the presence of block-wise missing data. In this paper we first investigate the situation of complete data and present a unified "bi-level" learning model for multi-source data. Then we give a natural extension of this model to the more challenging case with incomplete data. Our major contributions are three-fold: (1) the proposed models handle both feature-level and source-level analysis in a unified formulation and include several existing feature learning approaches as special cases; (2) the model for incomplete data avoids direct imputation of the missing elements and thus provides superior performances. Moreover, it can be easily generalized to other applications with block-wise missing data sources; (3) efficient optimization algorithms are presented for both the complete and incomplete models. We have performed comprehensive evaluations of the proposed models on the application of AD diagnosis. Our proposed models compare favorably against existing approaches.

Categories and Subject Descriptors

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

General Terms

Algorithms

Keywords

Alzheimer's disease, multi-source, block-wise missing data, optimization

1. INTRODUCTION

Recent advances in data collection technologies have made it possible to collect a large amount of data for many application domains. Very often, these data come from multiple sources. For instance, in the study of Alzheimer's Disease (AD), different types of measurements such as magnetic resonance imaging (MRI), positron emission tomography (PET), cerebrospinal fluid (CSF), blood test, protein expression data, and genetic data have been collected as they provide complementary information for the diagnosis of AD [31, 34]. In bioinformatics, different types of biological data including protein-protein interactions, gene expression and amino sequences have been collected for protein classification [19]. Extraction of the great wealth of information from such multi-source (a.k.a multi-modality) data has become a crucial step in knowledge discovery. Data mining and machine learning methods have been increasingly used to analyze multi-source data [26, 10, 29]. It is expected that the performance can be significantly improved if information from different sources can be properly integrated and leveraged. Multi-source learning has thus attracted great attentions in various application domains from biomedical informatics [17, 31] to web mining [1, 29]. It is closely related to multi-view learning. However they differ in several important aspects. More specifically, multi-view learning mainly focuses on semi-supervised learning and using unlabeled data to maximize the agreement between different views [2, 11]. In this paper, we focus on the multi-source learning in the supervised setting and we do not assume there are abundant unlabeled data available. In addition, we do not attempt to reduce the disagreement between multiple sources but try to extract complementary information from them, as is often the case in biomedical applications such as AD study.

In many applications, the data are also of very high dimension, e.g., medical images and gene/protein expression

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

ACM 978-1-4503-2174-7/13/08 ...$15.00.

data. However, the high-dimensional data often contains redundant information or even noisy or corrupted entries and thus poses a potential challenge. In order to build a stable and comprehensible learning model with good generalization, it is critical to perform certain "feature-pruning". A simple approach is to pool data from multiple sources together to create a single data matrix and apply traditional feature selection methods directly to the pooled data matrix. However, such an approach treats all sources equally important and ignores within-source and between-source relationship. Another popular approach is to adopt multiple kernel learning (MKL) to perform data fusion [19, 29, 31] which provides a principled method to perform source-level analysis, i.e., a particular source is considered relevant to the learning task only if its corresponding kernel is selected in MKL. However, MKL only performs source-level analysis, ignoring feature-level analysis. Such an approach is suboptimal when the individual data sources are high-dimensional and an interpretable model is desired. To fully take advantage of the multi-source data, it is desirable to build a model which performs both individual feature-level and source-level analysis. In this paper, we will use the term "bi-level analysis" (this was first introduced in [8]) to refer to the simultaneous feature-level and source-level analysis.

Besides the multi-modality and the high-dimensionality, the existence of (block-wise) missing data source is another big challenge encountered in most biomedical applications. Figure 1 provides a typical situation in AD research. We have 245 participants in total and 3 types of measurements (PET, MRI and CSF) are taken for diagnosis. Therefore for a single participant, there are at most three different measurements, which are represented in different colors. The blank region means that data from the corresponding source is missing. In this example, participants 1 $\sim$ 60 have records on PET and MRI but lack CSF information while participants 149 $\sim$ 245 have only MRI data. The block-wise missing data issue could emerge in several scenarios: inaccurate data sources of certain sample may be discarded; some data-collecting mechanisms (like PET) may be too costly to be applied to every participant; participants may not be willing to take certain measurements for various reasons. Notice that the missing data emerges in a block-wise way, i.e., for a patient, certain data source is either available or lost completely.

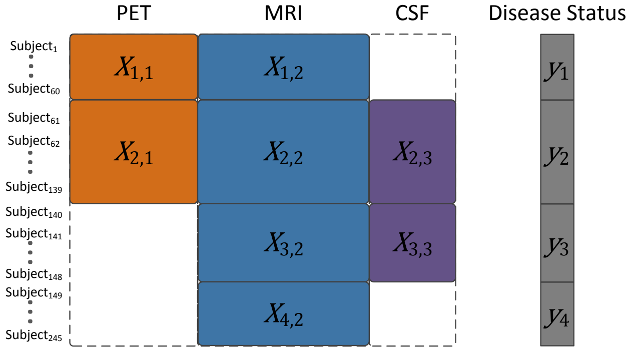

Figure 1: An illustration of an incomplete multi-source data with three sources.

1.1 Related Work

Considerable efforts have been made to deal with the missing data, both in data mining and biomedical informatics. Some well-known missing value estimation techniques like EM [12], iteratively singular value decomposition (SVD) and matrix completion [21] have been extended to biomedical applications by performing imputation on the missing part of the data. Although these approaches have demonstrated their effectiveness on handling random missing entries, they often deliver sub-optimal performance in AD research [32] for the following reasons: (1) these imputation approaches fail to capture the pattern of the missing data, i.e., the missing elements are not randomly scattered across the data matrix but emerge block-wisely. However, such prior knowledge is completely discarded in imputation methods; (2) due to the high-dimensionality of the data, these methods often have to estimate a significant amount of missing values, which would result in unstable performances.

To overcome the aforementioned drawbacks of standard imputation methods, Yuan et al. proposes an incomplete Multi-Source Feature learning method (iMSF) which avoids the direct imputation [32]. The iMSF method first partitions the patients into disjoint groups such that patients from the same group possess identical data source combinations. Feature learning is then carried out independently in each group and finally the results from all the groups are properly combined to obtain a consistent feature learning result. Such a mechanism enables iMSF to perform feature selection without estimating the missing values, however, the resulting model is unable to provide source-level analysis, i.e., we cannot tell which data source is more important for the diagnosis or which data source should be discarded in a particular application. Such a drawback may limit the performance of iMSF in applications where noisy or corrupted data sources are frequently encountered.

1.2 Main Contributions

Although the importance of bi-level analysis in bioinformatics has drawn increasing attention [8, 16, 28], how to effectively extend these techniques to deal with block-wise missing data remains largely unexplored. In this paper, we fill in such a gap by proposing a bi-level feature learning model for both complete and block-wise missing data. Our contributions are three-fold: (1) we propose a unified feature learning model for multi-source data, which includes several existing feature learning approaches as special cases; (2) we further extend this model to fit the block-wise missing data. The resulting model avoids direct imputation of the missing data and is capable of bi-level feature learning; (3) the proposed models for both of the complete and incomplete data require solving non-convex optimization problems. We present efficient optimization algorithms which find the solution by solving a sequence of convex sub-problems.

The rest of the paper is organized as follows. In Section 2, we present our unified framework for multi-source feature learning for complete data. The relationship between our model and existing works and the optimization algorithms are also discussed. In Section 3, we provide a natural extension of this model to deal with the block-wise missing data sources and propose an alternating minimization algorithm for the optimization. Extensive empirical evaluations are carried out in Section 4. We conclude the paper and point out some future directions in Section 5.

2. A UNIFIED FEATURE LEARNING MODEL FOR MULTI-SOURCE DATA

Assume we are given a collection of m samples from S data sources:

$$\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_S] \in \mathbb{R}^{m\times n}, \quad y \in \mathbb{R}^m,$$

where $\boldsymbol{X}_i \in \mathbb{R}^{m\times p_i}$ is the data matrix of the ith source with each sample being a p_i-dimensional vector, and $\boldsymbol{y}$ is the corresponding outcome for each sample. We consider the following linear model:

$$\boldsymbol{y} = \sum_{i=1}^{S} \boldsymbol{X}_i\boldsymbol{\beta}_i + \boldsymbol{\epsilon} = \boldsymbol{X}\boldsymbol{\beta} + \epsilon, \tag{1}$$

where each column of $\boldsymbol{X}$ is normalized to be zero mean and standard deviation of 1 and $\boldsymbol{\epsilon}$ represents the noise term. $\boldsymbol{\beta}$ is the underlying true model and is usually unknown in real-world applications. Based on $(\boldsymbol{X}, \boldsymbol{y})$, we want to learn an estimator of $\boldsymbol{\beta}$, denoted as $\hat{\boldsymbol{\beta}}$, whose non-zero elements $\mathcal{F} = \{j : \hat{\beta}_j \neq 0\}$ correspond to the relevant features. In other words, features correspond to the zero elements of $\hat{\boldsymbol{\beta}}$ are discarded. We consider the following regularization framework:

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \quad L(\boldsymbol{\beta}) + \Omega(\boldsymbol{\beta}),$$

where $L(\cdot)$ represents the data-fitting term and $\Omega(\cdot)$ is the regularization term which encodes our prior knowledge about $\boldsymbol{\beta}$. Specifically, the choice of $\Omega(\cdot)$ should also enable us to perform both feature-level and source-level analysis simultaneously. Towards this end, a natural approach is a two-stage model. First we learn different models for each data source and then combine these learned models properly. The regularization should be imposed independently on each stage to provide the bi-level analysis. We formalize our intuition as follows:

$$\underset{\boldsymbol{\alpha},\boldsymbol{\gamma}}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{y} - \sum_{i=1}^{S} \gamma_i \cdot \boldsymbol{X}_i\boldsymbol{\alpha}_i\|_2^2 + \sum_{i=1}^{S} \frac{\lambda_i}{p}\|\boldsymbol{\alpha}_i\|_p^p + \sum_{i=1}^{S} \frac{\eta_i}{q}|\gamma_i|^q, \tag{2}$$

where the minimization is taken with respect to $(\boldsymbol{\alpha}, \boldsymbol{\gamma})$ jointly. According to the intuition above, $\boldsymbol{\alpha}_i$ denotes the model learned on the ith data source and $\boldsymbol{\gamma}$ is the weight that combines those learned models together. The regularization is taken independently over $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ and therefore we have the flexibility to choose different values of p and q to induce sparsity on either feature-level or source-level. Notice that model (2) is not jointly convex and direct optimization towards (2) would be difficult. We provide an equivalent but simpler formulation in the following theorem and discuss its optimization in the next section.

THEOREM 1. *The formulation* (2) *is equivalent to the following optimization problem:*

$$\underset{\boldsymbol{\beta}}{\textit{minimize}} \ \frac{1}{2}\|\boldsymbol{y} - \sum_{i=1}^{S} \boldsymbol{X}_i\boldsymbol{\beta}_i\|_2^2 + \sum_{i=1}^{S} \nu_i\|\boldsymbol{\beta}_i\|_p^{\frac{pq}{p+q}}. \tag{3}$$

PROOF. Without loss of generality, we assume that $\boldsymbol{\alpha}_i \neq \boldsymbol{0}$ for all $i = 1, 2, \cdots, S$. Since if $\boldsymbol{\alpha}_i = \boldsymbol{0}$ for some i, the optimal γ_i must be 0 and therefore both $\boldsymbol{\alpha}_i$ and $\boldsymbol{\gamma}_i$ can be removed from (2). Let $\boldsymbol{\beta}_i = \gamma_i \cdot \boldsymbol{\alpha}_i$ and replace γ_i with $\frac{\|\boldsymbol{\beta}_i\|_p}{\|\boldsymbol{\alpha}_i\|_p}$, we can obtain an equivalent formulation:

$$\underset{\boldsymbol{\alpha},\boldsymbol{\beta}}{\text{minimize}} \ \frac{1}{2}\|\boldsymbol{y} - \sum_{i=1}^{S} \boldsymbol{X}_i\boldsymbol{\beta}_i\|_2^2 + \sum_{i=1}^{S} \frac{\lambda_i}{p}\|\boldsymbol{\alpha}_i\|_p^p + \sum_{i=1}^{S} \frac{\eta_i}{q}\left(\frac{\|\boldsymbol{\beta}_i\|_p}{\|\boldsymbol{\alpha}_i\|_p}\right)^q. \tag{4}$$

Taking partial derivative with respect to $\boldsymbol{\alpha}_i$ and setting it to zero leads to:

$$\eta_i\|\boldsymbol{\beta}_i\|_p^q = \lambda_i\|\boldsymbol{\alpha}_i\|_p^{p+q}, \quad i = 1, 2, \cdots, S. \tag{5}$$

Plugging (5) back into (4) with the change of variables, we get the formulation (3). □

2.1 Relation to previous works

Formulation (2) (or its equivalent form (3)) is a very general model. Assigning different values to p and q leads to various kinds of regularization and feature learning models. Next, we show several widely-used convex models are actually our special cases.

Let $p = 1$ and $q = \infty$. In this case, the regularization term in (3) becomes the ℓ_1-regularization and the resulting model becomes Lasso [25]:

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{y} - \sum_{i=1}^{S} \boldsymbol{X}_i\boldsymbol{\beta}_i\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1. \tag{6}$$

It is well-known that the ℓ_1-regularization leads to a sparse solution, which coincides with the goal of feature selection. However, it does not consider the source structure by treating all features from different sources equally.

On the other hand, if both p and q equal 2, then the ℓ_2-regularization is applied on each source. Letting $\nu_i = \lambda\sqrt{p_i}$ leads to the group lasso [33]:

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{y} - \sum_{i=1}^{S} \boldsymbol{X}_i\boldsymbol{\beta}_i\|_2^2 + \lambda\sum_{i=1}^{S} \sqrt{p_i}\|\boldsymbol{\beta}_i\|_2. \tag{7}$$

Similarly, if $p = \infty$ and $q = 1$, we obtain the $\ell_{1,\infty}$-regularization model [27, 23], which penalizes the largest elements of $\boldsymbol{\beta}_i$ for each source:

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{y} - \sum_{i=1}^{S} \boldsymbol{X}_i\boldsymbol{\beta}_i\|_2^2 + \sum_{i=1}^{S} \nu_i\|\boldsymbol{\beta}_i\|_\infty. \tag{8}$$

Besides these common convex formulations, our general model also includes a family of non-convex formulations which have not been fully explored in the literature. Particularly, letting $p = 1$ and $q = 2$ leads to the following non-convex model:

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{y} - \sum_{i=1}^{S} \boldsymbol{X}_i\boldsymbol{\beta}_i\|_2^2 + \sum_{i=1}^{S} \nu_i\|\boldsymbol{\beta}_i\|_1^{\frac{2}{3}}. \tag{9}$$

If $p = 2$ and $q = 1$, model (3) reduces to:

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{y} - \sum_{i=1}^{S} \boldsymbol{X}_i\boldsymbol{\beta}_i\|_2^2 + \sum_{i=1}^{S} \nu_i\|\boldsymbol{\beta}_i\|_2^{\frac{2}{3}}. \tag{10}$$

For the convex models such as lasso, the optimization algorithms have received intensive studies [5, 7, 13, 4]. In order to fully explore the functionality of our general model, we shall provide further investigations on the non-convex formulations in terms of optimization.

2.2 Optimization

We first focus on formulation (10), which is clearly a nonconvex optimization problem. Gasso et al. has shown in [15] that the ℓ_q-regularized least squares problem with $q < 1$ can be efficiently solved using the difference of convex functions (DC) algorithm [24]. The DC decomposition presented in [15] requires the regularization term to be a concave function with respect to the absolute value of the variable. However this is not the case in our formulation according to the following proposition:

Proposition 1. *Let*$f(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_2^{\frac{2}{3}}$. *Then f is neither convex nor concave w.r.t. $|\boldsymbol{\beta}|$ unless $\boldsymbol{\beta}$ is a scalar, where $|\cdot|$ denotes the absolute value.*

Proof. The proof is carried out by computing the Hessian of f. Without loss of generality, we assume $\boldsymbol{\beta} \neq \mathbf{0}$. It can be shown that:

$$\frac{\partial f}{\partial |\beta_i|} = \frac{2}{3}\|\boldsymbol{\beta}\|_2^{-\frac{4}{3}}|\beta_i|$$

$$\frac{\partial^2 f}{\partial |\beta_i| \partial |\beta_j|} = -\frac{8}{9}\|\boldsymbol{\beta}\|_2^{-\frac{10}{3}}|\beta_i \beta_j| + \mathbf{1}_{\{i=j\}} \cdot \frac{2}{3}\|\boldsymbol{\beta}\|_2^{-\frac{4}{3}},$$

where $\mathbf{1}$ is the indicator function. It is clear that, unless $\boldsymbol{\beta}$ is a scalar, in which case it is obvious that f is a concave function, $\frac{\partial^2 f}{\partial |\beta_i|^2}$ can be either positive or negative. In other words, the sign of the diagonal elements of the Hessian of f can be either positive or negative, which means that f is neither convex nor concave. □

To employ the DC algorithm, we need to avoid the nonconcavity of the regularization item. We introduce new variables $t_i, i = 1, 2, \cdots, S$ and transform (9) into the following formulation:

$$\begin{aligned} \underset{\boldsymbol{\beta}, \boldsymbol{t}}{\text{minimize}} \quad & \frac{1}{2}\|y - \sum_{i=1}^{S} \boldsymbol{X}_i \boldsymbol{\beta}_i\|_2^2 + \sum_{i=1}^{S} \nu_i t_i^{\frac{2}{3}} \\ \text{subject to} \quad & \|\boldsymbol{\beta}_i\|_2 \leq t_i, \quad i = 1, 2, \cdots, S. \end{aligned} \tag{11}$$

It is clear that (11) is equivalent to the original formulation (9), however the regularization term in (11) is concave with respect to t_i, as shown in Proposition 1. We apply the DC algorithm, i.e., for each $t_i^{\frac{2}{3}}$, we rewrite it as the difference of two convex functions as follows:

$$t_i^{\frac{2}{3}} = t_i - (t_i - t_i^{\frac{2}{3}}).$$

Therefore, (11) becomes:

$$\begin{aligned} \underset{\boldsymbol{\beta}, \boldsymbol{t}}{\text{minimize}} \quad & \frac{1}{2}\|y - \sum_{i=1}^{S} \boldsymbol{X}_i \boldsymbol{\beta}_i\|_2^2 + \sum_{i=1}^{S} \nu_i t_i - \sum_{i} \nu_i (t_i - t_i^{\frac{2}{3}}) \\ \text{subject to} \quad & \|\boldsymbol{\beta}_i\|_1 \leq t_i, \quad i = 1, 2, \cdots, S. \end{aligned} \tag{12}$$

Next we replace the second convex item $t_i - t_i^{\frac{2}{3}}$ by its affine minorant at the previous iteration. Specifically, suppose at the previous iteration the value of t_i is $\hat{t_i}$; now we approximate $t_i - t_i^{\frac{2}{3}}$ by its first-order Talyor expansion at $\hat{t_i}$ as follows:

$$(\hat{t_i} - \hat{t_i}^{\frac{2}{3}}) + (1 - \frac{2}{3}\hat{t_i}^{-\frac{1}{3}})(t_i - \hat{t_i}).$$

Plugging the above expression back to (12) and dropping the constant, we get:

$$\begin{aligned} \underset{\boldsymbol{\beta}, \boldsymbol{t}}{\text{minimize}} \quad & \frac{1}{2}\|y - \sum_{i=1}^{S} \boldsymbol{X}_i \boldsymbol{\beta}_i\|_2^2 + \sum_{i=1}^{S} \frac{2}{3}\widehat{t_i}^{-\frac{1}{3}} \nu_i t_i \\ \text{subject to} \quad & \|\boldsymbol{\beta}_i\|_2 \leq t_i, \quad i = 1, 2, \cdots, S. \end{aligned} \tag{13}$$

Since ν_i and $\hat{t_i}$ are nonnegative, all constraints in (13) must be active at the optimal points. Thus, (13) is equivalent to the following group lasso problem:

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \quad \frac{1}{2}\|y - \sum_{i=1}^{S} \boldsymbol{X}_i \boldsymbol{\beta}_i\|_2^2 + \sum_{i=1}^{S} \frac{2}{3}\widehat{t_i}^{-\frac{1}{3}} \nu_i \|\boldsymbol{\beta}_i\|_2.$$

After $\boldsymbol{\beta}$ is obtained, we update $\hat{t_i}$ with $\|\boldsymbol{\beta}_i\|_2$ and continue the iteration until convergence. Notice that $\hat{t_i}^{-\frac{1}{3}}$ can be very large if $\|\boldsymbol{\beta}_i\|_2$ is small. For numerical stability, we add a smoothing term θ to each $\widehat{t_i}$ as suggested by [15]. The overall procedure is summarized in Algorithm 1.

Algorithm 1 DC algorithm for solving (10)

```
Input: $\boldsymbol{X}$, $\boldsymbol{y}$, $\boldsymbol{\nu}$
Output: solution $\boldsymbol{\beta}$ to (10)
Initialize $\theta$, $\mu_i^{(0)}, i = 1, 2, \cdots, S$
for $k = 1, 2, \cdots$ do
    Update $\boldsymbol{\beta}$ and $\mu_i$ by:
        $\hat{\boldsymbol{\beta}}^k = \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^n} \frac{1}{2}\|\boldsymbol{y} - \sum_{i=1}^{S} \boldsymbol{X}_i \boldsymbol{\beta}_i\|_2^2 + \sum_{i=1}^{S} \mu_i^{k-1} \|\boldsymbol{\beta}_i\|_2$
        $\mu_i^k = \frac{2}{3}\nu_i(\|\hat{\boldsymbol{\beta}}_i^k\|_2 + \theta)^{-1/3}, \quad i = 1, 2, \cdots, S.$
    if the objective stops decreasing then
        return $\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}^k$
    end if
end for
```

Remark 1. *Model (9) can be solved in exactly the same way as above. The only difference is that in each iteration we need to solve a weighted lasso problem to get $\hat{\boldsymbol{\beta}}^{(\ell)}$.*

Remark 2. *Although we only consider the least squares loss function here, the above derivations can be easily extended to other widely-used convex loss functions, such as the logistic function.*

3. INCOMPLETE SOURCE-FEATURE SELECTION (ISFS) MODEL

In this section, we consider the more challenging and more realistic situation with block-wise missing data, as shown in Figure 1. In such situation, most patients do not have complete data collected from every data source but lack one or more data blocks. To apply existing feature learning approaches directly, we can either discard all samples that have missing entries or estimate the missing values based on the observed entries. However, the former approach may significantly reduce the size of the data set while the latter approach heavily relies on our prior knowledge about the missing values. Moreover, both approaches neglect the block-wise missing patterns in the data and therefore could lead to sub-optimal performance.

As in the case of complete data, an ideal model performs both feature-level and source-level analysis simultaneously. Next, we show how to extend the model on complete data presented in the previous section to a more general setting with missing data. Our intuition of designing such Incomplete Source-Feature Selection (iSFS) model is illustrated in Figure 2. We follow a similar strategy used in our complete model (2): individual model is learned on each data source and then all models are properly integrated via extra regularizations/constraints. As shown in Figure 2, we try to learn the model represented by $\boldsymbol{\beta}_1$, $\boldsymbol{\beta}_2$ and $\boldsymbol{\beta}_3$, corresponding to measurements from PET, MRI and CSF, respectively. A subtle issue is how to learn the coefficients $\boldsymbol{\alpha}$, since model (2) is not applicable due to the presence of missing data blocks. To address this issue, we partition the whole data set into multiple groups according to the availability of data sources, as illustrated in the red boxes in Figure 2. For this particular case, we partition the data into 4 groups, where the first group includes all the samples that have PET and MRI, the second group of patients possesses all three data sources, the third group of patients has MRI and CSF measurements, while the last group of patients only has MRI data. Note that within each group we have the complete data and the analysis from the previous section can be applied.

The proposed model is closely related to the iMSF model proposed in [32], however, they differ in several significant aspects: (1) the proposed method partitions the data into multiple groups according to the availability of data sources. The resulting groups are not disjoint compared to that of the iMSF. Generally, our partition method results in more samples for each group; (2) in the proposed approach, the model learned for each data source is consistent across different data source combinations while iMSF does not; (3) in every data source combination, we learn the weights of each source from the data. The weights for a specific data source may differ in different data source combinations. Unlike iMSF, the proposed method achieves source selection by discarding the data sources with a weight of 0. Thus, the proposed method is expected to outperform iMSF especially in the presence of noisy data sources.

3.1 Formulation

Before presenting the formal description of our iSFS model, we first introduce some notations which will simplify the discussion. Suppose we have S data sources in total and each participant has at least one data source available. Then there are $2^S - 1$ possible missing patterns: the number of all possible combinations of S data sources except for the case that all data sources are missing. For each participant, based on whether a certain data source is present, we obtain a binary indicator vector $I[1 \cdots S]$, where $I[i] = 1$ indicates the ith data source is available. For example in Figure 1, participants $1 \sim 139$ possess the same indicator vector $[1, 1, 0]$ while the indicator vector of participants $149 \sim 245$ is $[0, 1, 0]$. Using such indicator vectors simplifies our analysis. Moreover, we do not even need to store the complete vector for each participant but just need to record a single decimal integer if we convert this binary vector to a binary number, i.e., the information in the indicator vector can be completely described by a decimal integer, called **profile**. All these profiles are stored in an n-dimensional vector $pf[1 \cdots n]$ where n is the number of participants.

We are ready to give a concise description of our model. Following the aforementioned intuitions, we learn a consistent model (variable $\boldsymbol{\beta}$) across different source combinations, while within each combination, the weights (variable $\boldsymbol{\alpha}$) for different sources are learned adaptively. Mathematically, the proposed model solves the following formulation:

$$\begin{aligned} \underset{\boldsymbol{\alpha},\boldsymbol{\beta}}{\text{minimize}} \quad & \frac{1}{|\mathbf{pf}|} \sum_{m \in \mathbf{pf}} f(\boldsymbol{X}_m, \boldsymbol{\beta}, \boldsymbol{\alpha}_m, \boldsymbol{y}_m) + \lambda \mathbf{R}_{\boldsymbol{\beta}}(\boldsymbol{\beta}) \\ \text{subject to} \quad & \mathbf{R}_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}_m) \leq 1 \quad \forall m \in \mathbf{pf}, \end{aligned} \tag{14}$$

where

$$f(\boldsymbol{X}, \boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{y}) = \frac{1}{n} \mathbf{L}(\sum_{i=1}^{S} \alpha^i \boldsymbol{X}^i \boldsymbol{\beta}^i, \boldsymbol{y}) \tag{15}$$

and $\mathbf{R}_\alpha$, $\mathbf{R}_\beta$ are regularizations on $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ respectively. The m subscript in (14) denotes the matrix/vector restricted to the samples that contain m in their profiles. $\boldsymbol{X}^i$ and $\boldsymbol{\beta}^i$ in (15) represent the data matrix and and the model of the ith source, respectively. $\mathbf{L}$ can be any convex loss function such as the least squares loss function or the logistic loss function and n is number of rows of $\boldsymbol{X}$.

3.2 Optimization

One of the advantages of iMSF is its efficient optimization algorithm. In fact, iMSF can be solved by standard convex multi-task learning algorithms [3, 20]. The proposed iSFS model involves a more complicated optimization problem. In fact, (14) is not jointly-convex w.r.t $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, posing a major challenge. We adapt the alternating minimization method to solve (14). More specifically, we first initialize $\boldsymbol{\beta}$ and compute the optimal $\boldsymbol{\alpha}$. Then $\boldsymbol{\beta}$ is updated based on the computed $\boldsymbol{\alpha}$. We keep this iterative procedure until convergence. For simplicity, we focus on the least squares loss function in the following discussion. The techniques can be easily extended to other loss functions, e.g., the logistic loss function.

3.2.1 *Computing $\boldsymbol{\alpha}$ when $\boldsymbol{\beta}$ is fixed*

As shown in Figure 2, we learn the weight $\boldsymbol{\alpha}$ for each source combination independently. Therefore, when $\boldsymbol{\beta}$ is fixed, the objective function of (14) is decoupled w.r.t $\boldsymbol{\alpha}_m$ and the optimal $\boldsymbol{\alpha}_m$ is given by the optimal solution of the following problem:

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\text{minimize}} \quad & \|\sum_{i=1}^{S} \alpha^i \boldsymbol{X}^i \boldsymbol{\beta}^i - \boldsymbol{y}\|_2^2 \\ \text{subject to} \quad & \mathbf{R}_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}) \leq 1. \end{aligned} \tag{16}$$

For many choices of the regularization term $\mathbf{R}_{\boldsymbol{\alpha}}$, such as the ridge penalty, the ℓ_1-norm penalty as well as other sparsity-induced penalties [4], the optimal solution of (16) can be efficiently computed via the accelerated gradient algorithm [6].

3.2.2 *Computing $\boldsymbol{\beta}$ when $\boldsymbol{\alpha}$ is fixed*

When we keep $\boldsymbol{\alpha}$ fixed and seek the optimal $\boldsymbol{\beta}$, (14) becomes an unconstrained regularization problem:

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \quad g(\boldsymbol{\beta}) + \lambda \mathbf{R}_{\boldsymbol{\beta}}(\boldsymbol{\beta}) \tag{17}$$

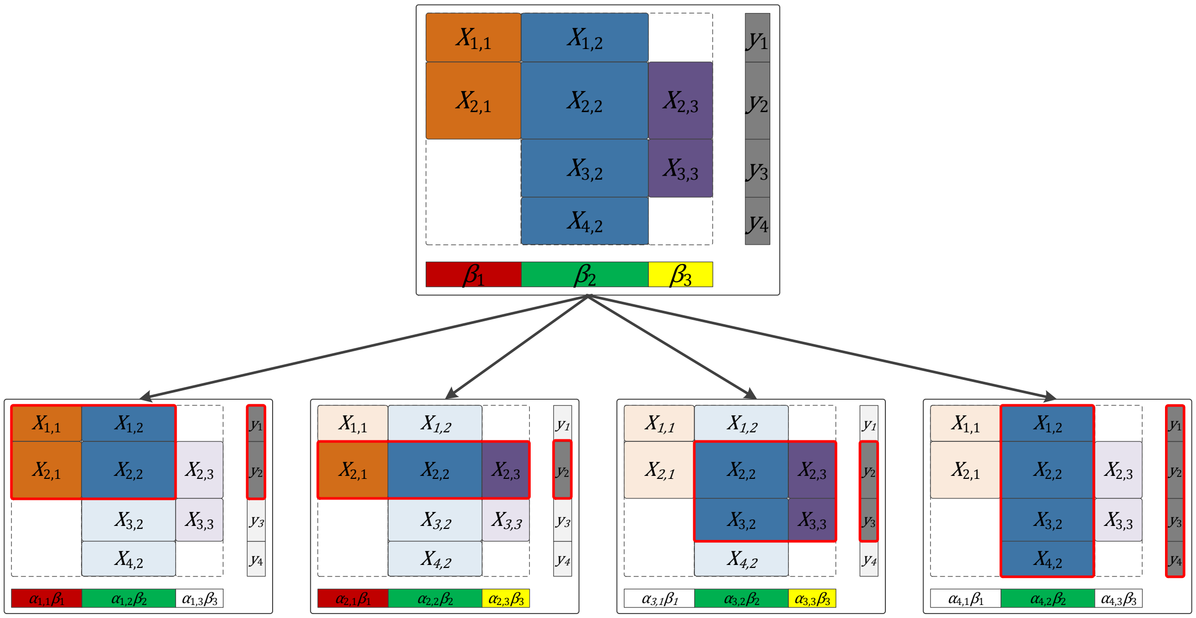

Figure 2: Illustration of the proposed learning model. The data set is partitioned into four groups according to the availability of data sources, as highlighted by the red boxes. The goal is to learn three models $\boldsymbol{\beta}_1$, $\boldsymbol{\beta}_2$ and $\boldsymbol{\beta}_3$ for each data source as well as the coefficient $\boldsymbol{\alpha}$ that combines them. Notice that, for the ith data source, $\boldsymbol{\beta}_i$ remains identical while $\boldsymbol{\alpha}$ may vary across different groups.

where

$$g(\boldsymbol{\beta}) = \frac{1}{|\mathbf{pf}|}\sum_{m\in\mathbf{pf}}\frac{1}{2n_m}\|\sum_{i=1}^{S}(\boldsymbol{\alpha}_m^i\boldsymbol{X}_m^i)\boldsymbol{\beta}_m^i - \boldsymbol{y}_m\|_2^2.$$

and n_m is number of rows of $\boldsymbol{X}_m$. We can observe that $g(\boldsymbol{\beta})$ is a quadratic function of $\boldsymbol{\beta}$ and thus the overall formulation is to minimize the summation of a quadratic term and a regularization term: a typical formulation that can be solved efficiently via accelerated gradient method provided that the following proximal operator [9]:

$$\underset{\boldsymbol{\beta}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{\beta} - \boldsymbol{v}\|_2^2 + \lambda\mathbf{R}_\beta(\boldsymbol{\beta})$$

can be computed efficiently. Indeed, this is the case for many widely used regularization terms. In addition, in order to apply standard first-order lasso solvers, we only need to provide the gradient of $\boldsymbol{\beta}$ at any given point without knowing the explicit quadratic form. For each data source i, we can compute the gradient of the $g(\boldsymbol{\beta})$ w.r.t $\boldsymbol{\beta}^i$ as follows:

$$\begin{aligned}\nabla g(\boldsymbol{\beta}^i) = &\frac{1}{|\mathbf{pf}|}\sum_{m\in\mathbf{pf}}\frac{1}{n_m}\mathbf{I}(m \;\&\; 2^{S-i} \neq 0)\\ &(\boldsymbol{\alpha}_m^i\boldsymbol{X}_m^i)^T(\sum_{i=1}^{S}\boldsymbol{\alpha}_m^i\boldsymbol{X}_m^i\boldsymbol{\beta}_m^i - \boldsymbol{y}_m),\end{aligned} \tag{18}$$

where $\mathbf{I}(\cdot)$ is the indicator function which equals 1 when the condition is satisfied and 0 otherwise. The expression $m \;\&\; 2^{S-i} \neq 0$ ensures that the ith source exists in the combination m, where & denotes the bit-wise AND operation. Then we can obtain $\nabla g(\boldsymbol{\beta})$ by stacking all the $\nabla g(\boldsymbol{\beta}^i)$, $i = 1, 2, \cdots S$ and finally obtain a global solution of (17) via applying the accelerated gradient method. Algorithm 2 summarizes our alternating minimization scheme.

Algorithm 2 Iterative algorithm for solving (14)

```
Input: $\boldsymbol{X}$, $\boldsymbol{y}$, $\lambda$
Output: solution $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$ to (14)
Initialize $(\boldsymbol{\beta}^i)^0$ by fitting each source individually on the available data.
for $k = 1, 2, \cdots$ do
    Compute each $(\boldsymbol{\alpha})^k$ via solving a constrained lasso problem (16).
    Update $(\boldsymbol{\beta})^k$ via solving a regularized lasso problem (17).
    if the objective stops decreasing then
        return $\boldsymbol{\beta} = (\boldsymbol{\beta})^k$
    end if
end for
```

REMARK 3. *Our model can be easily extended to the logistic loss function which is widely used in classification problems. Computing $\boldsymbol{\alpha}$ in (16) amounts to solving a constrained logistic regression problem while computing $\boldsymbol{\beta}$ in (17) requires solving a regularized logistic regression problem. In fact, any convex loss function can be applied to our model as long as the gradient information can be efficiently obtained.*

REMARK 4. *We may apply different forms of $\mathbf{R}_\alpha$ and $\mathbf{R}_\beta$ in order to capture more complex structures, as long as the associated proximal operator can be efficiently computed. Particularly, we can employ the ℓ_1-norm penalty to achieve simultaneous feature-level and source-level selection.*

REMARK 5. *A special case of the proposed iSFS model can be obtained by setting $\boldsymbol{\alpha}_m$ to $\frac{1}{n_m}$ for every m, where n_m is the number of samples that have profile m. As a result, the optimization (14) only involves $\boldsymbol{\beta}$ and becomes a convex* *programming problem. In fact, this is exactly an extension of the classical lasso method to the block-wise missing data. To the best of our knowledge, such an extension is not known in existing literature.*

4. EXPERIMENTS

To examine the efficacy of the proposed bi-level feature learning models, we report the performance of the proposed models for the complete and block-wise missing data, on both synthetic data and real-world applications. Specifically, the following aspects are evaluated: (i) model (9) and (10) for complete data; (ii) model (14) for block-wise missing data; (iii) the capability of source-level analysis.

4.1 Comparison on complete data

We first evaluate the effectiveness of the complete models (9) and (10) on synthetic data generated by the linear model (1). The parameter settings follow the similar strategy described in [14, 30]. Specifically, we have $S = 20$ sources in total and the underlying true model $\boldsymbol{\beta} = [\boldsymbol{\beta}_1^T, \boldsymbol{\beta}_2^T, \cdots, \boldsymbol{\beta}_S^T]^T$ only takes non-zero values in the first six sources, whose values are 10, 8, 6, 4, 2 and 1 respectively. The data matrix $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_S]$ and the noise term $\boldsymbol{\epsilon}$ all follow the Gaussian distribution with zero mean and standard deviation of 0.5. To evaluate the performance of bi-level feature learning, we consider the following two situations: (1) all features within the six sources are useful, i.e., the elements of $\boldsymbol{\beta}_i$, $i = 1, 2, \cdots, 6$ are all non-zero; (2) not all features within the six sources are useful, i.e., $\boldsymbol{\beta}_i$ is sparse for $\boldsymbol{\beta}_i$, $i = 1, 2, \cdots, 6$. Specifically, only the first 3 features within each $\boldsymbol{\beta}_i$ are nonzero. Figure 3 illustrates these two settings:

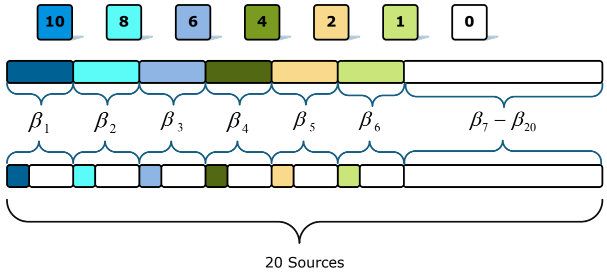

Figure 3: Two scenarios of the underlying true model $\boldsymbol{\beta}$: the upper one corresponds to the situation of non-sparse features and the lower one represents the situation of sparse features. The white block represents zero elements while the non-zero values are represented by different colors, indicated in the first row.

For each scenario, we partition the dataset into disjoint training set and test set, and compare models (9) and (10) with lasso, group lasso and sparse group lasso. 5-fold cross-validation is employed to tune the parameters for each model. Specifically, the set of tuning parameters for lasso, group lasso, model (9) and model (10) are chosen from the interval $M = [10^{-8}, 10^2]$. For sparse group lasso, its parameters are chosen from the product space of $M \times M$. We report the number of features and groups selected by each model and the mean squared error (MSE) on the testing set. In addition, since we know the underlying true model $\boldsymbol{\beta}$, we also include the parameter estimation error: $\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2^2$, where $\hat{\boldsymbol{\beta}}$ is the estimated model. All the results are averaged over 10 replications and are listed in Table 1. For simplicity, we use FRAC$(1, 2)$ to denote model (9) ($p = 1, q = 2$) and FRAC$(2, 1)$ to denote model (10). The experimental results show that, in the situation of sparse features, model (9) achieves the least MSE and estimation error, while for the non-sparse feature case, model (10) outperforms the others. In addition, in both cases, models (9) and (10) demonstrate significant improvement over the lasso, group Lasso and sparse group lasso.

4.2 Comparison on block-wise missing data

Next we consider the more realistic setting where block-wise missing data is present. We evaluate our models using the classification of Alzheimer's disease. We utilize the Alzheimer's Disease Neuroimaging Initiative (ADNI) data set [22, 18] and choose 4 data sources for each patient: Proteomics, PET, MRI and CSF. We investigate the classification between AD patient, normal control (NC) subjects, stable MCI subjects (non-converter) and progressive MCI subjects (converter). Imputation methods such as Mean-value imputation, EM, KNN, iterative SVD and matrix completion as well as the iMSF feature learning model are included for comparison. Notice that kernel learning algorithms are not applicable here since the data are incomplete. All the evaluations are done in a two-stage fashion. In the first stage, we either apply the feature learning methods to select informative features or the imputation methods to fill in the missing entries in the data. Then in the second stage, the Random Forest classifier is applied to perform the classification. We use 10% of the ADNI data for training and report the accuracy, sensitivity, specificity and the area under the ROC curve (AUC value) on the remaining test data. 5-fold cross-validation is used for selecting suitable parameters for iSFS, iMSF, KNN and SVD. Particularly, for iSFS, iMSF and matrix completion, we choose five values from $[10^{-5}, 10]$ in the log scale as candidates. For KNN, the size of the neighborhood is selected from $[1, 5, 10, 15, 20, 25]$. The rank parameter in the SVD is chosen from $[5, 10, 15, 20, 25, 30]$. In addition, we employ the ℓ_1-norm penalty for both $\mathbf{R}_\alpha$ and $\mathbf{R}_\beta$. The results are presented in Table 2 to Table 4. All the results are averaged over 10 repetitions. From the evaluation results, we can observe that: (1) among all imputation methods, the mean-value imputation and EM demonstrate better performance in terms of accuracy. However, their results are not stable, as revealed by the low sensitivity/specificity value in some tasks; (2) feature learning models such as iSFS and iMSF provide superior results than the imputation methods and often achieve uniform improvement across all the measurements. This coincides with our intuition that estimating the missing blocks directly is usually difficult and unstable and approaches avoiding imputation are more preferred. In particular, iSFS clearly delivers the best performance among all approaches.

4.3 Capability of source selection

Motivated by the strategies used in [19], we add two random (noisy) data sources to the ADNI data set to verify the performance of source-level learning. We compare our iSFS model with iMSF and report their performance in Figure 4. Besides the previous tasks, two additional evaluations: AD

Table 1: Performance on synthetic complete data. The MSE denotes the mean squared error of prediction on the test set and ESTI stands for the parameter estimation error. For the scenario of sparse feature, the underling true model has 6 groups and 18 features, while for the situation of non-sparse feature, the true model takes 6 groups and 60 features. All results are averaged over 10 replications.

Methods	Sparse features				Non-sparse features			
	MSE	Esti	# Group	# Feature	MSE	Esti	# Group	# Feature
frac(1, 2)	**17.04**	**15.36**	**6.1**	**30.6**	1618.47	1245.87	12.6	94.0
frac(2, 1)	146.15	131.04	6.4	64.0	**242.27**	**221.01**	**5.1**	**51.0**
lasso	256.84	257.47	17.1	71.7	2007.61	1617.81	19.3	141.3
group lasso	165.55	162.35	13.5	135.0	669.80	493.23	12.4	124.0
sparse group glasso	71.69	80.93	13.1	77.9	729.22	552.79	13.8	137.9

Table 2: Classification results of AD patients and normal controls. All results are averaged over 10 replications.

	Accuracy	Sensitivity	Specificity	AUC
iSFS	**0.8103**	**0.8077**	0.8124	**0.8101**
iMSF	0.7857	0.7671	0.8005	0.7838
SVD	0.7756	0.7770	0.7746	0.7758
KNN	0.7668	0.7161	0.8072	0.7617
Mean	0.7789	0.7845	0.7744	0.7795
EM	0.8089	0.7963	**0.8189**	0.8076
MC	0.5957	0.5710	0.6155	0.5932

Table 3: Classification results of AD patients and stable MCI patients. All results are averaged over 10 replications.

	Accuracy	Sensitivity	Specificity	AUC
iSFS	**0.7489**	**0.7032**	0.7816	**0.7424**
iMSF	0.7172	0.6910	0.7359	0.7135
SVD	0.6942	0.6510	0.7250	0.6880
KNN	0.6774	0.6819	0.6742	0.6781
Mean	0.7338	0.6163	**0.8177**	0.7170
EM	0.7174	0.6323	0.7782	0.7052
MC	0.6234	0.6135	0.6304	0.6220

Table 4: Classification results of progressive MCI patients and normal controls. All results are averaged over 10 replications.

	Accuracy	Sensitivity	Specificity	AUC
iSFS	**0.8754**	0.9361	**0.8297**	**0.8829**
iMSF	0.8611	0.9190	0.8174	0.8682
SVD	0.7280	0.7222	0.7323	0.7273
KNN	0.7272	0.6381	0.7944	0.7162
Mean	0.7889	**0.9531**	0.6651	0.8091
EM	0.8027	0.8281	0.7836	0.8059
MC	0.7740	0.7728	0.7749	0.7738

patients vs. MCI and MCI vs. normal controls, are also included. We can see that our method outperforms the iMSF model in most of the cases. Such a result again justifies the importance of source-level analysis when noisy/corrupted data sources are present.

5. CONCLUSION

In this paper, we investigate the bi-level feature learning motivated by biomedical applications and propose systematic approaches for both complete and block-wise missing data. Specifically, we introduce a unified feature learning model for complete data, which contains several classical convex models as special cases. We further show that the model for complete data can be easily extended to handling the more challenging block-wise missing data. The effectiveness of the proposed models are verified through both synthetic data and the Alzheimer's disease study.

In future work, we plan to apply the proposed algorithms to other applications involving block-wise missing data. In addition, we plan to analyze the generalization performance of the proposed algorithms.

6. REFERENCES

[1] A. Aizawa and K. Oyama. A fast linkage detection scheme for multi-source information integration. In *Web Information Retrieval and Integration, 2005. WIRI'05. Proceedings. International Workshop on Challenges in*, pages 30–39. IEEE, 2005.

[2] R. Ando and T. Zhang. Two-view feature generation model for semi-supervised learning. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, pages 25–32, 2007.

[3] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

[4] F. Bach. Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*, 4(1):1–106, 2011.

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

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

[7] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge university press, 2004.

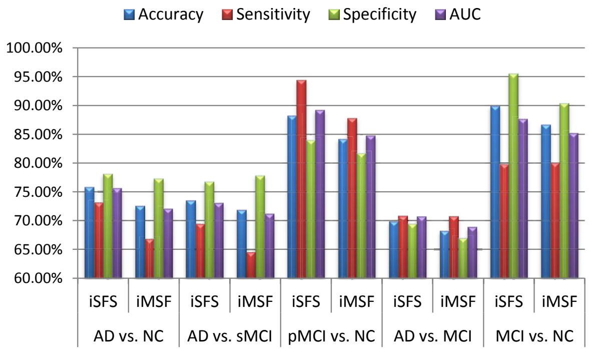

Figure 4: The classification results of iSFS and iMSF on ADNI data set with additional noisy data sources.

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

[9] P. Combettes and J. Pesquet. Proximal splitting methods in signal processing. *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, 2010.

[10] K. Crammer, M. Kearns, and J. Wortman. Learning from multiple sources. *The Journal of Machine Learning Research*, 9:1757–1774, 2008.

[11] M. Culp, G. Michailidis, and K. Johnson. On Multi-view learning with additive models. *The Annals of Applied Statistics*, 3(1):292–318, 2009.

[12] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. 1997.

[13] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

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

[15] G. Gasso, A. Rakotomamonjy, and S. Canu. Recovering sparse signals with a certain family of nonconvex penalties and DC programming. *Signal Processing, IEEE Transactions on*, 57(12):4686–4698, 2009.

[16] J. Huang, P. Breheny, and S. Ma. A selective review of group selection in high dimensional models. *arXiv preprint arXiv:1204.6491*, 2012.

[17] I. Huopaniemi, T. Suvitaival, J. Nikkilä, M. Orešič, and S. Kaski. Multivariate multi-way analysis of multi-source data. *Bioinformatics*, 26(12):i391–i398, 2010.

[18] 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. *Journal of Magnetic Resonance Imaging*, 27(4):685–691, 2008.

[19] G. Lanckriet, T. De Bie, N. Cristianini, M. Jordan, and W. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, 2004.

[20] J. Liu, S. Ji, and J. Ye. Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 339–348. AUAI Press, 2009.

[21] R. Mazumder, T. Hastie, and R. Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11:2287–2322, 2010.

[22] S. G. Mueller, M. W. Weiner, L. J. Thal, R. C. Petersen, C. Jack, W. Jagust, J. Q. Trojanowski, A. W. Toga, and L. Beckett. The alzheimerąŕs disease neuroimaging initiative. *Neuroimaging Clinics of North America*, 15(4):869, 2005.

[23] A. Quattoni, X. Carreras, M. Collins, and T. Darrell. An efficient projection for $\ell_{1,\infty}$ regularization. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 857–864, 2009.

[24] P. Tao and L. An. Convex analysis approach to dc programming: Theory, algorithms and applications. *Acta Math. Vietnam*, 22(1):289–355, 1997.

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

[26] O. Troyanskaya, K. Dolinski, A. Owen, R. Altman, and D. Botstein. A bayesian framework for combining heterogeneous data sources for gene function prediction (in saccharomyces cerevisiae). *Proceedings of the National Academy of Sciences*, 100(14):8348, 2003.

[27] B. Turlach, W. Venables, and S. Wright. Simultaneous variable selection. *Technometrics*, 47(3):349–363, 2005.

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

[29] Z. Xu, I. King, and M. Lyu. Web page classification with heterogeneous data fusion. In *Proceedings of the 16th international conference on World Wide Web*, pages 1171–1172. ACM, 2007.

[30] H. Yang, Z. Xu, I. King, and M. Lyu. Online learning for group lasso. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 2010.

[31] J. Ye, K. Chen, T. Wu, J. Li, Z. Zhao, R. Patel, M. Bae, R. Janardan, H. Liu, G. Alexander, et al. Heterogeneous data fusion for alzheimer's disease study. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1025–1033. ACM, 2008.

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

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

[34] D. Zhang and D. Shen. Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in alzheimer's disease. *NeuroImage*, 59(2):895–907, 2012.