Diagnostic Sparse Connectivity Networks with Regularization Template

Yue Qu, Chuanren Liu*, Kai Zhang, Keli Xiao, Bo Jin, Hui Xiong

Abstract—Real-world dynamic systems and complex objects are often monitored with multivariate time series where each dimension represents a system signal. Performing accurate diagnostic for a group of dynamic systems while simultaneously taking into account their similarities/distinctions, is a non-trivial task. In this paper, we develop an adaptive regularization approach to learning sparse connectivity structures in complex dynamic systems. The learned connectivity networks shed lights on the structural compositions of the system and hence can serve as highly informative inputs for various machine learning tasks. In particular, we focus on high-dimensional and semi-supervised learning scenarios and present a joint learning method to recover system-wise connectivity patterns by adaptively constructing a shared, sparsity-inducing regularization template across all systems. The shared template can be intuitively interpreted and used as a modeling template for efficient analysis of new systems. Moreover, our approach can flexibly incorporate information such as must-links and cannot-links for constructing regularization templates. Overall, our approach, named *sparse adaptive regularization* (SAR), can extract structure-related connectivity features efficiently and effectively, and result in significant improvements for machine learning tasks in dynamic systems. We benchmark our approach against the state-of-the-art methods with real-world data. Our results demonstrate the superiority of our approach over the baselines in terms of accuracy, efficiency, and interpretability.

Index Terms—Dynamic System, Sparse Network, Adaptive LASSO, Shared Regularization.

1 Introduction

Many real-world objects can be deemed dynamic systems. For instance, studies in healthcare informatics often use data collected by monitoring patients (as systems) with different lab tests and health indicators (as system signals). In this way, each system is measured with a multivariate time series where each dimension represents a system signal. Performing accurate diagnostic analytics for a group of such dynamic systems is an important and non-trivial task. In this paper, we develop an effective and efficient approach for learning hidden connectivity networks associated with dynamic systems, so that various machine learning tasks, such as classification of the systems, can be accurately and conveniently implemented.

There have been pilot studies analyzing the multivariate time series collected from dynamic systems for a variety of diagnostic purposes. For instance, the connectivity network in the complex brain system measured with fMRI (functional magnetic resonance imaging) data has been used to diagnosis neuropsychiatric disorders [13, 16] and mild cognitive impairment [28]. Similarly, large-scale connectivity network is used to model the patients measured by a large number of BOLD (blood-oxygen level dependent) signals [31]. Intuitively, the connectivity between signals in dynamic systems may bear important information and can serve as informative inputs for further analytical tasks.

When the dynamic systems are measured with high dimensional data, a sparse regularization on the connectivity network is necessary to avoid overfitting as well as to improve interpretability of the modeling results. For example, connectivity networks can be constructed by thresholding the connectivity weights (e.g., correlations between two time series features) to localize the focal regions of high connectivity. Instead of simple thresholding, Peng et al. [21] used sparse regression models and the LASSO (least absolute shrinkage and selection operator) [24] regularization to compute sparse partial correlation or inverse covariance as the connectivity weights in the network.

However, existing approaches mainly recover the connectivity network for each individual system independently and in an unsupervised way. In contrast, it is of great value to identify the common network structure shared by many systems in a cohort (e.g., a patient population). For example, the common network structure shared by many patients may bear meaningful information that can inspire medicine researchers for novel investigations and development of new treatments. Moreover, there are often many types of noises in the data associated with individual systems, so that the independent network construction can be inaccurate and even erroneous. A joint network construction approach with a shared network structure can leverage the data from multiple systems to improve the construction performances. Moreover, the network construction process is expected to be able to incorporate supervising information which can further improve learning performances for applications

- *Yue Qu and Bo Jin are with Dalian University of Technology, Dalian, China. Email: quyue1541@mail.dlut.edu.cn, jinbo@dlut.edu.cn.*
- *Chuanren Liu is with the University of Tennessee, Knoxville, TN, USA. Email: cliu89@utk.edu.*
- *Kai Zhang is with East China Normal University, Shanghai, China. Email: kzhang980@gmail.com.*
- *Keli Xiao is with Stony Brook University, Stony Brook, NY, USA. Email: Keli.Xiao@stonybrook.edu.*
- *Hui Xiong is with Rutgers University, Newark, NJ, USA. Email: hxiong@rutgers.edu.*
- ** Corresponding author.*

such as predictive modeling with the connectivity networks.

Our work aims to address the aforementioned challenges. Specifically, we propose a novel sparse regularization formulation that construct connectivity networks of all systems in a collaborative way. In addition to the connectivity networks, the approach will also optimize a common sparse structure shared by all systems. The common structure can be intuitively interpreted and used as a regularization template in constructing the networks for new systems. We also show that our approach can effectively leverage supervising information such as must-links and cannot-links between the systems. Since system networks are constructed in a collaborative process, the supervising information also influences the shared regularization template. Once constructed, we use the network structures of the systems as inputs for further machine learning tasks. To the best of our knowledge, this is the first work that introduces a shared sparse regularization template for collaboratively learning connectivity network structures in multiple dynamic systems.

To sum up, our main contributions are:

- We develop a new approach for obtaining sparse connectivity networks in dynamic systems, leading to superior performances in diagnostic tasks (such as classification) for the systems.
- In estimating network structures for a cohort of dynamic systems, our method can not only jointly construct the connectivity networks for all systems but also learn a shared sparse network structure as the regularization template.
- The constructed system networks can be used for a variety of analytical tasks for dynamic systems. In this paper, we use the networks for predictive modeling in several applications with real-world data. Extensive experimental results show improved learning performances in terms of accuracy, efficiency, and interpretabilty of connectivity networks.

The remainder of this paper is organized as follows. We discuss related work in Section 2. We discuss problem settings and review existing algorithms in Section 3. The details of our methodology are developed in Section 4, and the implementation and algorithm details are provided in Section 5. We then introduce the data sets and the evaluation metrics in Section 6 and discuss the experimental results in Section 7. Finally we conclude the paper in Section 8.

2 Related Work

Sparse connectivity networks has attracted much attention in various application areas. For instance, Du et al. [9] proposed a sparse canonical correlation analysis (SCCA) model with the novel Absolute GraphNet (AGN) penalty to identify genetic markers and associations. As another example in computational biology, Wang et al. [26] exploited the function category correlations for protein function prediction. In particular, there is a series of research on sparse network inference based on correlation analysis for a cohort of dynamic systems. The key idea is to construct connectivity networks by measuring the correlation structures in the systems. Candès and Wakin [7] suggested that LASSO [24, 5] is an effective method to ensure the sparsity of the connectivity networks which can still capture rich information of the dynamic systems. Many recent studies have attempted to design the sparse constrains and/or regularizers in learning the sparse connectivity structures effectively and efficiently [21, 31, 28, 1]. Essentially, these methods are developed to solve the multivariate time series classification problem, which has been conventionally approached using nearest neighbor (NN) classifiers coupled with different distance functions, such as the Dynamic Time Warping (DTW) [18], for the multivariate time series data. The research on sparse connectivity networks takes a new perspective by representing each multivariate time series with a network.

Particularly, connectivity networks have been adopted to address various challenges in data-driven biological and medical research. For example, Peng et al. [21] used the sparse partial correlation estimation for selecting nonzero partial correlations with sparse regression techniques. The approach was applied to identify hub genes with a microarray breast cancer data set. Huang et al. [13] studied the sparse inverse covariance estimation (SICE), also known as the exploratory Gaussian graphical model, for modeling brain connectivity networks. Lee et al. [16] used LASSO to model sparse brain networks from a small number of noisy measurements for autism studies. Zhang et al. [31] identified two connectivity measures that were shown to be informative in pattern classification tasks with epileptic patients. Also using LASSO, Wee et al. [28] considered a sparse linear regression model to construct brain networks. Specifically, the ℓ_1-norm penalization is used to ensure sparsity in the networks, and the ℓ_2-norm penalization is used to ensure consistent non-zero connections across multiple systems (i.e., patient/control subjects). Huang et al. [12] proposed a sparse simplex model for brain anatomical and genetic network analysis, where the model can address the shift-invariant and parameter tuning problems to reveal the brain spatial expression patterns. In Bayesian fashion, Adametz and Roth [1] proposed the Translation-invariant Matrix-T process (TiMT) to recover the covariance networks in studying the biological pathways in cancer patients.

Different from existing methods, which seldom leverage the shared information of all the dynamic systems, our method aims to obtain a regularization template which can be shared by all systems in constructing connectivity networks. The template is motivated by the idea of using adaptive LASSO for simultaneous estimation and variable selection [32]. The purpose is to overcome the problem that LASSO variable selection which can be inconsistent in some conditions [19, 17]. By sharing information among multiple dynamic systems, the template in our framework can improve not only the modeling performance of the sparse connectivity networks but also the computational efficiency in processing new system instances.

Connectivity networks of dynamic systems are useful for various data mining and machine learning tasks (e.g., clustering, classification, and regression). For example, we may exploit the network structures with graph-based learning/embedding algorithms, which are trending research topics in recent years. Also, we can apply the graph convolutional neural networks proposed by Kipf and Welling [14] for link prediction and graph classification tasks with

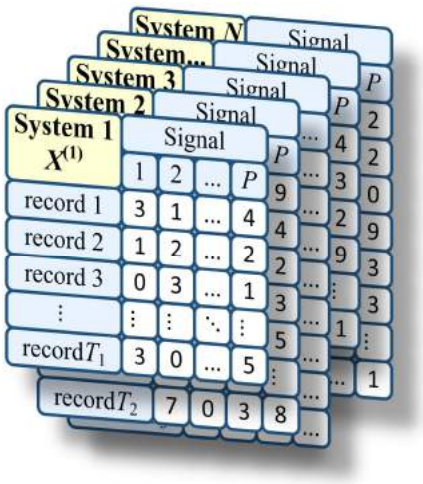


Fig. 1: The multivariate time series matrices of N dynamic systems. Each matrix $X^{(n)}$ has the same signal dimension P, while its number of records T_n might be different.

the connectivity networks. With the extension by Veličković et al. [25], the connectivity networks can also be used for unsupervised learning tasks, such as dynamic system clustering with the multi-head attention mechanism. More recently, Gao et al. [10] proposed the learnable graph convolutional model, which can automatically select neighboring nodes for each feature and transform the networks into grid-like structures for more general convolutional operations.

3 Preliminaries

The general structure of a group of dynamic systems is demonstrated in Figure 1, where each system is associated with a multivariate time series. Suppose the n-th dynamic system is monitored by multivariate time series $X^{(n)} \in \mathbb{R}^{T_n \times P}$, for $n = 1, 2, \cdots, N$. The multivariate dimension of $X^{(n)}$ is P, which corresponds to the number of dynamic signals (e.g., fMRI sensors, medical measures, etc.) used to measure the systems. The number of observations in $X^{(n)}$, T_n, can be different for different systems. As aforementioned, the 'system' can stand for many dynamic subjects in real-world applications, such as a machine or a patient.

To construct connectivity network $\beta^{(n)}$ for the n-th system, our method is based on least absolute shrinkage and selection operator (LASSO). Introduced by Tibshirani [24] and Breiman [5], LASSO performs variable selection via sparse regularization, to enhance the prediction accuracy and interpretability of the statistical model it produces. This section reviews the state-of-the-art approaches using LASSO technique for constructing connectivity network structures.

Notations: We use $\|\cdot\|_F$ as the matrix Frobenius norm. Without causing ambiguity, we define the dot-product of two vectors x, y as: $\langle x, y\rangle = x'y$ and the trace-product of two matrices X, Y as: $\langle X, Y\rangle = \mathrm{tr}(X'Y)$. We use $\circ$ as the operator of element-wise product of two matrices or vectors. The function $diag(X)$ returns the vector of diagonal values in matrix X, while $diag(x)$ returns a diagonal matrix with diagonal values from the vector x.

3.1 Connectivity Network for Dynamic Systems

One simple way to construct connectivity network for dynamic system is to compute the correlation matrix [11]:

$$\beta_{ij}^{(n)} = \rho\left(X_{*i}^{(n)}, X_{*j}^{(n)}\right), \tag{1}$$

where $X_{*i}^{(n)}$ is the i-th column in the matrix $X^{(n)}$, and ρ is the correlation measure. If Pearson correlation coefficient (PCC) is the correlation measure, it follows that $\beta^{(n)} = \langle X^{(n)}, X^{(n)}\rangle = (X^{(n)})'X^{(n)}$ assuming that all dynamic signals are centered and normalized. The PCC can be further transformed to enhance its normality, such as the Fisher transform [27]:

$$\beta_{ij}^{(n)} = \frac{1}{2}\ln\left(\frac{1+\rho_{ij}^{(n)}}{1-\rho_{ij}^{(n)}}\right), \tag{2}$$

where $\rho_{ij}^{(n)} = PCC\left(X_{*i}^{(n)}, X_{*j}^{(n)}\right)$. Alternatively, we can compute partial correlation instead of PCC as one signal may correlate to multiple signals in the same system. Examples using partial correlation include the empirical association networks by computing the pseudo inverse (PINV) of the covariance matrix [23] and the sparse extension [2]. We will include these benchmark methods in our experiments.

For systems of large scale, sparse connectivity networks can result in better modeling performance and easier interpretation [21, 16, 2]. The general formulation of sparse network construction can be written as:

$$\min_{\beta} \sum_{n} \frac{1}{2}\left\|X^{(n)}\beta^{(n)} - X^{(n)}\right\|_F^2 + \Omega(\beta), \tag{3}$$

where $\beta^{(n)} \in \mathbb{R}^{P\times P}$ is the constructed network. We let $\beta_{ii}^{(n)} = 0, \forall n, i$, and thus each dynamic signal of one system is regressed with other signals in the system. We assume all systems are measured with the same set of signals, and the signals in $X^{(n)}$ have been standardized, such that $X_{*j}^{(n)}$ has mean of zero and standard deviation of one. Figure 2 shows one example of sparse connectivity network.

The regularization $\Omega(\beta)$ can be used to represent assumptions from different scientific backgrounds, and has been implemented with various approaches in the literature. In the following we summarize previous methods using the regularization $\Omega(\beta)$ to enforce sparsity on the connectivity networks of dynamic systems.

3.1.1 Sparse Regularization

One example of the regularization $\Omega(\beta)$ is the sparse regularization proposed by Tibshirani [24] and Lee et al. [16]:

$$\Omega(\beta) = \lambda \sum_{n}\sum_{ij}\left|\beta_{ij}^{(n)}\right|. \tag{4}$$

The sparse regularization uses the parameter λ to control the sparsity of the connectivity networks. However, this regularization allows us to decompose the overall optimization problem in Equation 3 to a set of independent problems for different instances n and signals j as:

$$\min_{\beta_{*j}^{(n)}} \frac{1}{2}\left\|X^{(n)}\beta_{*j}^{(n)} - X_{*j}^{(n)}\right\|_2^2 + \lambda\sum_{i}\left|\beta_{ij}^{(n)}\right|. \tag{5}$$

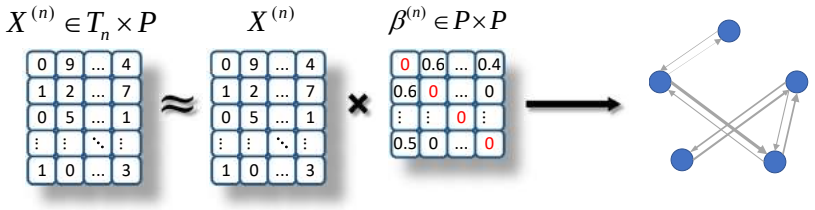


Fig. 2: Example of the sparse connectivity network.

Thus there is no structural information shared by different instances and signals. In comparison, we will jointly construct the connectivity networks for all instances with a shared regularization template. After the learning process, the resultant regularization template can be used to efficiently model new system instances.

3.1.2 Re-weighted LASSO Regularization

To enhance the modeling performance with sparsity patterns, Candes et al. [8] proposed the re-weighted LASSO regularization. The idea is to incorporate the 'weighted' regularization in the connectivity networks:

$$\min_{\beta^{(n)}} \frac{1}{2} \left\| X^{(n)} \beta^{(n)} - X^{(n)} \right\|_F^2 + \sum_{ij} \lambda_{ij}^{(n)} \left| \beta_{ij}^{(n)} \right| \tag{6}$$

where $\lambda_{ij}^{(n)} > 0$ is the weight for the coeffecient $\beta_{ij}^{(n)}$. The motivation is that, if the weights $\lambda_{ij}^{(n)}$ are set wisely, it could improve the modeling performance. Specifically, Candes et al. [8] proposed the 'iterative re-weighting' to update the weights at each iteration with the previous solution:

$$\lambda_{ij}^{(n)} = \frac{1}{\left| \beta_{ij}^{(n)} \right| + \epsilon}, \tag{7}$$

where ϵ is a constant. Such iterative process is equivalent to:

$$\min_{\beta^{(n)}} \frac{1}{2} \left\| X^{(n)} \beta^{(n)} - X^{(n)} \right\|_F^2 + \sum_{ij} \log \left(\left| \beta_{ij}^{(n)} \right| + \epsilon \right). \tag{8}$$

Nevertheless, the re-weighted LASSO regularization cannot share structural information among multiple networks and cannot learn a shared regularization template to estimate future networks.

3.1.3 Mixed Norm Regularization

There exist pilot studies to jointly construct a group of connectivity networks. For example, Wee et al. [28] adopted the regularization by a mixed norm to promote group based sparsity by keeping the network topology to be identical among instances, while at the same time allowing variation under the regularization. Specifically, the mixed norm regularization is formulated as:

$$\Omega(\beta) = \lambda \sum_{ij} \sqrt{\sum_n \left(\beta_{ij}^{(n)} \right)^2}. \tag{9}$$

However, this Mixed Norm Regularization (MNR) approach cannot return a regularization template. According to Equation 9, when a new instance of dynamic system is to be included in the analysis, we need to recompute the connectivity networks for $N+1$ instances, which can be time consuming and limit the practical usage of the connectivity networks for real-world applications. To address this issue, our approach will optimize a regularization template, which can be used to model new system instances efficiently.

3.1.4 Weighted Graph Regularization

Recently, Yu et al. [30] proposed the weighted graph regularization (WGraphSR) to generate the connectivity networks, with the optimization objective:

$$\begin{aligned} &\min_{\beta^{(n)}} \frac{1}{2} \left\| X^{(n)} \beta^{(n)} - X^{(n)} \right\|_F^2 + \\ &\frac{\lambda_1}{2} \sum_{ij} P_{ij}^{(n)} \left\| \beta_{*i}^{(n)} - \beta_{*j}^{(n)} \right\|_2^2 + \lambda_2 \sum_{ij} C_{ij}^{(n)} \left| \beta_{ij}^{(n)} \right|, \end{aligned} \tag{10}$$

where,

$$P_{ij}^{(n)} = PCC\left(X_{*i}^{(n)}, X_{*j}^{(n)} \right),$$
$$C_{ij}^{(n)} = \exp\left(-\frac{\left(P_{ij}^{(n)} \right)^2}{\sigma} \right),$$

and $\lambda_1, \lambda_2, \sigma$ are hyper-parameters. The first regularization term penalizes $\left\| \beta_{*i}^{(n)} - \beta_{*j}^{(n)} \right\|_2^2$ with a weighted graph where the edge weights are the Pearson correlation $P_{ij}^{(n)}$. The second regularization term uses more penalty for $\left| \beta_{ij}^{(n)} \right|$ if $C_{ij}^{(n)}$ is larger (i.e., $P_{ij}^{(n)}$ is smaller). In other words, the connectivity coefficients of highly correlated system signals are regularized to be similar and the connectivity networks will be sparse between system signals with low correlations. However, Equation 10 does not consider multiple systems jointly with shared regularization structure.

4 METHODOLOGY

In comparison with the above methods, we develop a *shared adaptive regularization* (SAR) framework, which can jointly construct sparse networks with a shared regularization template, which, in turn, is learned in the optimization process. Our work is based on the adaptive LASSO, which was analyzed by Zou [32]. Specifically, we formulate the adaptive sparse regularization for the joint construction of the connectivity networks for all the instances $\{X^{(n)} : n = 1, \cdots, N\}$ by minimizing the following objective function:

$$\sum_n \frac{1}{2} \left\| X^{(n)} \beta^{(n)} - X^{(n)} \right\|_F^2 + \sum_n \sum_{ij} w_{ij}^{-\gamma} \left| \beta_{ij}^{(n)} \right| + \lambda I(\beta), \tag{11}$$

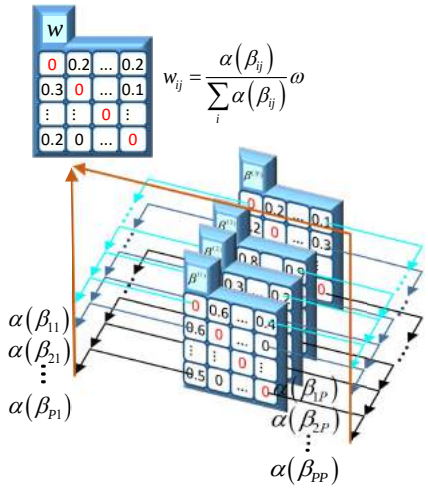


Fig. 3: Example of the shared regularization template.

where $\beta_{ii}^{(n)} = 0, \forall n, i$. The *regularization template* $w \in \mathbb{R}^{P\times P}$ is shared among all instances in the adaptive LASSO regularization. To avoid trivial solutions, we have constrains $w_{ij} \geq 0$, $\forall i, j$, and $\sum_i w_{ij} = \omega$, $\forall j$. In addition, the last term $I(\beta)$ can encode more information, e.g., a set of must-links and cannot-links on the network instances. We present details in the following two subsections.

4.1 The Shared Regularization Template

We propose two methods to compute the template w, while the two procedures can be written with the same general formula. In the general formula, we compute the weight $w_{ij} = \alpha(\beta_{ij})$ where $\alpha(\cdot)$ is an aggregate function of the coefficients $\beta_{ij}^{(n)}$ for $n = 1, \cdots, N$. With the constraint $\sum_i w_{ij} = \omega$, we can rewrite the weight as following:

$$w_{ij} = \frac{\alpha(\beta_{ij})}{\sum_i \alpha(\beta_{ij})}\omega. \tag{12}$$

The aggregating and weighting procedures to compute the template are demonstrated in Figure 3.

4.1.1 Harmonic Template

For the first method, we use the harmonic mean as the aggregate function:

$$\alpha(\beta_{ij}) = \frac{N}{\sum_n 1/\left|\beta_{ij}^{(n)}\right|}. \tag{13}$$

Intuitively, this leads to the group sparse effect that we have $\beta_{ij}^{(n)} \to 0$ for all $n \neq n'$ when we have some n' such that $\beta_{ij}^{(n')} \to 0$. To see this group sparse effect, note the fact that $\alpha(\beta_{ij}) \to 0$ when we have $\beta_{ij}^{(n')} \to 0$. This fact in turn implies that $w_{ij} \to 0$ and the coefficients $\beta_{ij}^{(n)}$ will be regularized to be close to zero.

4.1.2 Arithmetic Template

For the second method, we identify the arithmetic template w by minimizing Equation 11. That is, we solve w to:

$$\min_w \sum_n \frac{1}{2}\left\|X^{(n)}\beta^{(n)} - X^{(n)}\right\|_F^2 + \sum_n \sum_{ij} w_{ij}^{-\gamma}\left|\beta_{ij}^{(n)}\right| + \lambda I(\beta).$$

The solution is given in the following theorem.

Theorem 1. *Define the aggregate function $\alpha(\cdot)$ based on the arithmetic mean*

$$\alpha(\beta_{ij}) = \left(\frac{1}{N}\sum_n \left|\beta_{ij}^{(n)}\right|\right)^{1/(\gamma+1)}, \tag{14}$$

Then the arithmetic template can be computed with Equation 12.

Proof. The arithmetic template problem is indeed to:

$$\min_w \frac{1}{N\gamma}\sum_{ij} w_{ij}^{-\gamma}\sum_n \left|\beta_{ij}^{(n)}\right|, \tag{15}$$

subject to:

$$w_{ij} \geq 0, \forall i, j;$$
$$\sum_i w_{ij} = \omega, \forall j.$$

Note that, the factor $\frac{1}{N\gamma}$ helps to reduce the complexity of derivatives without changing the solution. By introducing the Lagrangian multiplier η_j for constraint $\sum_i w_{ij} = \omega$, we have the following Lagrangian function to minimize:

$$\mathcal{L}(w, \eta) = \frac{1}{N\gamma}\sum_{ij} w_{ij}^{-\gamma}\sum_n \left|\beta_{ij}^{(n)}\right| + \sum_j \eta_j\left(\sum_i w_{ij} - \omega\right). \tag{16}$$

By solving the KKT optimal conditions $\nabla_{w_{ij}}\mathcal{L}(w, \eta) = 0$, the only feasible solution is:

$$w_{ij} = \left(\frac{\frac{1}{N}\sum_n \left|\beta_{ij}^{(n)}\right|}{\eta_j}\right)^{1/(\gamma+1)} = \frac{\alpha(\beta_{ij})}{\eta_j^{1/(\gamma+1)}}. \tag{17}$$

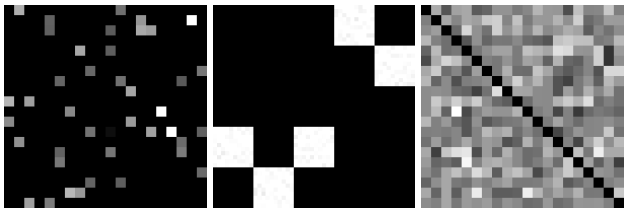
Since $\sum_i w_{ij} = \omega$, it follows that:

$$\eta_j^{1/(\gamma+1)} = \frac{\sum_i \alpha(\beta_{ij})}{\omega},$$
$$w_{ij} = \frac{\alpha(\beta_{ij})}{\sum_i \alpha(\beta_{ij})}\omega.$$

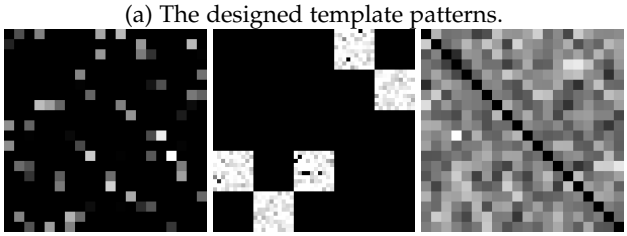
□

4.1.3 Sanity Check and Discussions

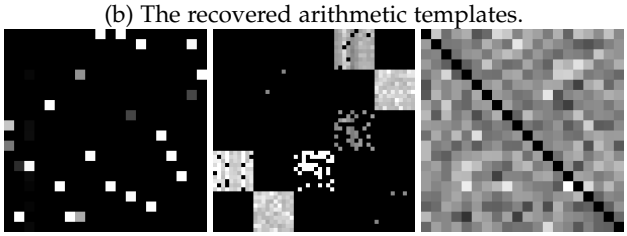
Note that, we can always compute the template matrix, which is shared in constructing all networks. However, when there is no shared sparsity pattern among the networks, the values in the template matrix would be (approximately) constant. To show this and demonstrate the feasibility to identify the shared sparsity pattern, we compute the shared regularization template with some simulated data sets by controlling the existence of shared sparsity patterns. Specifically, we generate N sparse matrices $\beta_o^{(n)}$ and their corresponding dynamic systems $X^{(n)}$, such that $X^{(n)}\beta_o^{(n)} \approx X^{(n)}$, for $n = 1, 2, \cdots, N$. Then, we use Equation 11 (with $\lambda = 0$) to compute the connectivity networks $\beta^{(n)}$ and the shared (arithmetic and harmonic) regularization template w. Finally, we compare the computed template w and the 'real' template w_o directly computed with the original $\beta_o^{(n)}$, $n = 1, 2, \cdots, N$.



(a) The designed template patterns.



(b) The recovered arithmetic templates.



(c) The recovered harmonic templates.

Fig. 4: The templates designed in the generated networks and the templates recovered by our methods.

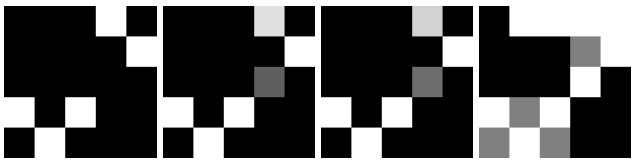


Fig. 5: The first plot is the designed sparsity pattern in 5-dimensional systems; the second and third plots are arithmetic (computed in 0.36 seconds) and harmonic (computed in 0.07 seconds) templates identified by our approach, respectively; and the last plot is based on the Bayesian optimization (computed in 257.38 seconds) without constraints.

By comparing the templates designed in the generated networks and the templates computed with our methods, Figure 4 shows that our approach can successfully recover the designed patterns. Specifically, the first two cases (columns) are designed with shared sparsity patterns, which are both recovered by our approach. The second case (column) using the shared block-wise sparsity pattern is especially easy to discern. Note that, the third case (column) is designed not to follow a shared sparsity pattern, and the computed template is roughly uniform as expected.

To investigate if the shared sparsity patterns can be recovered without constraints used in our arithmetic and harmonic templates, we also implement the Bayesian optimization (BO) to optimize templates[1]. However, we have to significantly reduce the system dimension to compute Bayesian optimization quickly. In Figure 5, the comparison shows that the pattern recovery with Bayesian optimization can take much longer time yet result in less accurate recovery performance. These observations suggest that the template optimization without constraints is challenging, and our arithmetic and harmonic templates enable efficient optimization of the regularization templates that lead to good learning performance.

Note that we can obtain the template based on either the harmonic or the arithmetic way, as they both can result in effective regularization on network sparsity. In this rest of the paper, we use the name SAR to refer to our approach with the arithmetic template, and use the name SAR-harm for harmonic template. Once computed, the regularization template stores the shared information among all instances, and the template can be used to efficiently construct connectivity networks for new instances.

1. We use the MATLAB function: https://www.mathworks.com/help/stats/bayesopt.html

4.2 The Link Constraints

The incorporation of side-information with semi-supervised learning has been adopted widely in the literature [3, 29, 15]. In this paper, to encode must-links and cannot-links with $I(\beta)$, we let $e_{pq} = 1$ if there is a must-link between the p-th system and the q-th system, otherwise we let $e_{pq} = -1$ if we have a cannot-link. Then we minimize $I(\beta)$ defined as the sum of the squared link residuals:

$$I(\beta) = \sum_{(pq)} \frac{1}{2} \left\| \beta^{(p)} - e_{pq}\beta^{(q)} \right\|_F^2 . \tag{18}$$

Intuitively, we want $\beta^{(p)}$ and $\beta^{(q)}$ to be close if there is a must-link between instances p and q. In the case of a cannot-link, we want $\beta^{(p)}$ and $\beta^{(q)}$ to be opposite.

Equivalently, the link information can be represented with an incidence matrix $R \in \mathbb{R}^{C\times N}$ where C is the number of links and N is the number of instances. For the c-th row representing the link between instance p and instance q, we let $R_{cp} = 1$ and $R_{cq} = -e_{pq}$. With the incidence matrix defined, we have:

Theorem 2. *The regularization term $I(\beta)$ can be represented as:*

$$I(\beta) = \sum_{n_1 n_2} \frac{1}{2} L_{n_1 n_2} \langle \beta^{(n_1)}, \beta^{(n_2)} \rangle. \tag{19}$$

where $L = R'R$.

Proof. By definition we have:

$$\begin{aligned} \sum_{(pq)} \|\epsilon_{pq}\|_F^2 &= \sum_c \left\| \sum_n R_{cn}\beta^{(n)} \right\|_F^2 \\ &= \sum_c \sum_{n_1 n_2} R_{cn_1} R_{cn_2} \langle \beta^{(n_1)}, \beta^{(n_2)} \rangle \\ &= \sum_{n_1 n_2} L_{n_1 n_2} \langle \beta^{(n_1)}, \beta^{(n_2)} \rangle \end{aligned}$$

Thus, it follows that:

$$I(\beta) = \sum_{n_1 n_2} \frac{1}{2} L_{n_1 n_2} \langle \beta^{(n_1)}, \beta^{(n_2)} \rangle.$$

□

The matrix L is positive semidefinite, so it can be deemed the graph Laplacian defined with the link information. It is also straightforward to show that, if $i \neq j$, then:

$$L_{ij} = \begin{cases} -1 & \text{A must-link between } i \text{ and } j \\ 1 & \text{A cannot-link between } i \text{ and } j \\ 0 & \text{No link between } i \text{ and } j \end{cases} \quad (20)$$

If $i = j$, the diagonal L_{ii} is the number of links containing i, i.e., $L_{ii} = \sum_{j \neq i} |L_{ij}|$.

An illustrative demonstration of the must/cannot-links are shown in Figure 6. Incorporating such link information can lead to better performances for further learning tasks with the connectivity networks. In practice, we may create must-links for instances with the same class label and assign cannot-links for those with different class labels.

5 Implementation and Algorithm Details

Since we have two sets of variables (β, w) in Equation 11, we optimize each of them iteratively. First, when β is fixed, we can update w with one of two procedures proposed in Subsection 4.1. Second, when w is fixed, we need to optimize β in a constrained quadratic programming problem with adaptive LASSO regularizations. Since there are N systems, we update $\beta^{(n)}$ in block-wise for $n = 1, 2, \cdots, N$. With all other blocks fixed, we have the problem to:

$$\min_{\beta^{(n)}} \frac{1}{2} \left\| X^{(n)} \beta^{(n)} - X^{(n)} \right\|_F^2 + \sum_{ij} w_{ij}^{-\gamma} \left| \beta_{ij}^{(n)} \right| + \lambda I \left(\beta^{(n)} \right), \quad (21)$$

where

$$I \left(\beta^{(n)} \right) = \frac{1}{2} L_{nn} \left\| \beta^{(n)} \right\|_F^2 + \sum_{n' \neq n} L_{nn'} \langle \beta^{(n')}, \beta^{(n)} \rangle. \quad (22)$$

The convergence of this block wise optimization procedure can be guaranteed since the matrix L is positive semidefinite. We develop the parallel shrinkage updating algorithm to optimize $\beta^{(n)}$ in the following subsection.

5.1 The Parallel Shrinkage Updating Algorithm

By reorganizing the quadratic terms and the linear terms, Equation 21 is equivalent to:

$$\min_{\beta^{(n)}} \frac{1}{2} \mathrm{tr}(\beta^{(n)'} Q^{(n)} \beta^{(n)}) - \langle c^{(n)}, \beta^{(n)} \rangle + \sum_{ij} w_{ij}^{-\gamma} \left| \beta_{ij}^{(n)} \right|, \quad (23)$$

where

$$\begin{aligned} Q^{(n)} &= X^{(n)'} X^{(n)} + \lambda L_{nn} I, \\ c^{(n)} &= X^{(n)'} X^{(n)} - \lambda \sum_{n' \neq n} L_{nn'} \beta^{(n')}. \end{aligned} \quad (24)$$

For better efficiency, the parallel shrinkage updating algorithm consists of two components: parallel updating and shrinkage operator.

Algorithm 1 The gradient descent with shrinkage operator.

1: **repeat**
2: $\quad \beta_{*j}^{(n)} \leftarrow \beta_{*j}^{(n)} - h \cdot \left(Q^{(n)} \beta_{*j}^{(n)} - c_{*j}^{(n)} \right)$
3: $\quad \beta_{*j}^{(n)} \leftarrow shrink_{h \cdot w_{*j}^{-\gamma}} \left(\beta_{*j}^{(n)} \right)$
4: **until** Convergence

Algorithm 2 The overall algorithm.

1: Compute L, $Q^{(n)}$, and $c^{(n)}$, $\forall n$.
2: Initialize $\beta^{(n)}$, $\forall n$.
3: **repeat**
4: $\quad$ Compute the harmonic or arithmetic template w.
5: $\quad$ **for** $n = 1, 2, \cdots, N$ **do**
6: $\quad\quad$ **for** $j = 1, 2, \cdots, P$ **do**
7: $\quad\quad\quad$ Run Algorithm 1 in parallel to update $\beta_{*j}^{(n)}$.
8: $\quad\quad$ **end for**
9: $\quad$ **end for**
10: **until** Convergence

5.1.1 *Parallel updating.*

We first observe that the columns $\beta_{*j}^{(n)}$, $\forall j$, can be optimized independently when updating the $\beta^{(n)}$. This observation suggests that we may update the columns in parallel for the best of computing efficiency. Specifically, the optimization problem for updating $\beta_{*j}^{(n)}$ is:

$$\min_{\beta_{*j}^{(n)}} \frac{1}{2} \beta_{*j}^{(n)'} Q^{(n)} \beta_{*j}^{(n)} - c_{*j}^{(n)'} \beta_{*j}^{(n)} + \sum_i w_{ij}^{-\gamma} \left| \beta_{ij}^{(n)} \right|. \quad (25)$$

It shows that the updating of $\beta_{*j}^{(n)}$ is a LASSO regularized quadratic programming problem.

5.1.2 *Shrinkage operator.*

Following Beck and Teboulle [4], Parikh et al. [20], we apply the widely-used proximal gradient descent with the shrinkage operator to optimize Equation 25 with the LASSO regularization. The shrinkage operator is defined as:

$$shrink_\eta(u) = sign(u) \circ (\|u\| - \eta)_+, \quad (26)$$

where η is a vector of positive values, $sign(u)$ is the signs in vector u, and $(u)_+$ replaces negative values in vector u with zeros. With these definitions, the algorithm to optimize Equation 25 is presented in Algorithm 1, where the step size h should be dynamically updated to ensure convergence.

5.1.3 *Diagonal constraints.*

Note that our constraints $\beta_{jj}^{(n)} = 0$, $\forall n, j$, can be effectively satisfied by: (i) initializing $\beta_{jj}^{(n)} = 0$, $\forall n, j$; (ii) setting *always* the diagonal of $c^{(n)}$ to be zero, $\forall n$; (iii) setting *temporally* the j-th row and column of $Q^{(n)}$ to be zero for computing $\beta_{*j}^{(n)}$.

5.2 The Overall Algorithm.

The overall algorithm is presented in Algorithm 2, which can compute both the template w and the connectivity networks $\beta^{(n)}$ for all systems $n = 1, 2, \cdots, N$. In practice, given a large number of dynamic systems from which the shared regularization template is already learned, the

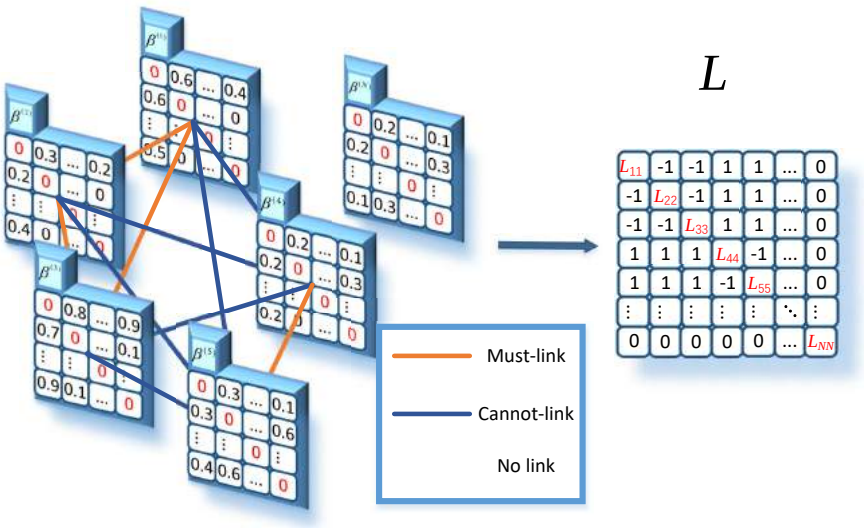


Fig. 6: Example of the link constraints.

template can be directly used in constructing the network for a new system, or constructing networks for many new systems in parallel.

6 Experimental Settings

To evaluate the performances of our algorithms, we introduce three application data sets, the evaluation metrics, and the benchmark methods.

6.1 Application Data Sets

We evaluate our approach by applying it to practical applications that cover the topics from disease diagnosis and human activity detection:

The Parkinson's Progression Markers Initiative (PPMI)[2] investigates patient cohorts of significant interest using 212 features from heterogeneous sources including advanced imaging, biologic sampling and clinical and behavioral assessments to identify the conditions of Parkinson's disease (PD) progression. Data classes include: PD and non-PD.

Parkinson Speech Dataset (PS)[3] contains 48 PD patients and 20 healthy individuals who appealed at the Department of Neurology in Cerrahpasa Faculty of Medicine, Istanbul University [22]. A group of 26 sound recordings features such as sustained vowels, numbers, words and short sentences are extracted from each voice sample of each individual. Data classes include: PD and non-PD.

Occupancy Detection Dataset (OD)[4] has been used for binary classification tasks to detect room occupancy based on multiple features (e.g., temperature, humidity, light, and CO2) [6]. Ground-truth occupancy was obtained from time stamped pictures that were taken every minute. Data classes include: occupied and not occupied.

Table 1 summarizes important data statistics. Particularly, we are interested in the relationship between the quality of the connectivity networks and the ratio between the system dimension and the average record size. Therefore, we define a proxy p/r ratio as follows:

$$p/r = P \Big/ \frac{\sum_{n=1}^{N} T_n}{N}.$$

For instance, PPMI has $p/r \approx 9$, the highest in our data sets. The ratio value has a wide range in our experiments.

6.2 Evaluation Metrics

To validate our approach, we construct connectivity networks for all instances and then conduct classification tasks. The classification results are used to measure the performance of the obtained connectivity networks. Specifically, for evaluating the classification performance, we use the following evaluation metrics:

$$\begin{aligned} Accuracy &= \frac{TP + TN}{P + N} \\ Precision &= \frac{TP}{TP + FP} \\ Recall &= \frac{TP}{TP + FN} \\ F1 &= \frac{2 \times TP}{2 \times TP + FP + FN} \end{aligned}$$

where P and N are the number of positive (case) subjects and negative (control) subjects, respectively. Also, TP, TN, FP, and FN are the number of true positives, true negatives, false positives, and false negatives. The AUC score (Area under the ROC Curve) is considered as well.

For all the evaluation metrics, we report the average and standard deviation obtained from the randomized 5-fold cross validation. In addition, the degree of sparsity and the computational cost are also important metrics evaluating the effectiveness of the approaches in extracting useful information. Note that, we use only the connectivity networks for evaluating the classification performance. Incorporating additional system features may result in further improvements in real-world applications.

2. http://www.ppmi-info.org/
3. http://archive.ics.uci.edu/ml/
4. http://archive.ics.uci.edu/ml/

TABLE 1: Summary of data sets

Data	Instance Size	Record Size	Feature Size	Average Record Size	Number of Classes	p/r
PPMI	683	15798	212	23.1303	2	9.17
PS	68	1208	26	17.7647	2	1.46
OD	115	20560	5	178.7826	2	0.03

6.3 Benchmark Methods

We compare our approach, the *Shared Adaptive Regularization* (SAR), with various benchmarks, including PCC [11], PCC+Fisher [27], PINV [23], SR-C [2], SR-PC [16], RW-LASSO [8], MNR [28], and WGraphSR [30]. To test the importance of the link constraints, we also implement the Shared Adaptive Regularization without the link constraints (SAR-noLC). That is, we set $\lambda = 0$ in Equation 11. For all methods, we first obtain connectivity networks and then conduct classification tasks with Logistic Regression (LR), Support Vector Machine (SVM), and Fully Connected Neural Network (FCNN) applied on the connectivity networks. The linear kernel has been chosen for SVM, since the connectivity networks are usually high-dimensional and preliminary tests show that the linear kernel is more efficient and accurate for our data. The FCNN has two hidden layers, each of which has 100 neurons. Note that, all methods are evaluated with cross validation and all parameters are tuned with respect to accuracy.

7 Results and Analysis

This section discusses the experimental results regarding the classification performance, computational efficiency, along with the visualization of sparse networks.

7.1 Classification Performance

Table 2 shows the performance comparison based on the PPMI data. The best results are shown in bold. As can be seen, SR-PC, RW-LASSO, MNR, and our SAR methods all achieve good sparsity and classification results, while other methods do not perform well. Similarly as shown in Table 3 and Table 4, our approach can achieve the best or the second best performance. Interestingly, the results of the RW-LASSO approach often have the best sparsity.

Overall, the results suggest that: (1) The classic methods (PCC, PCC+Fisher, and PINV) lead to the unsatisfactory performance in terms of the classification accuracy as well as the sparsity. This is expected because these methods cannot consider the sparsity regularization, and the constructed dense connectivity networks would result in difficulties for further machine learning tasks, such as classification. (2) Both SR-C and SR-PC rely on the sparse regularization technique, so they can construct sparse networks, while SR-PC has a better classification performance. These results are consistent with the claims by Lee et al. [16], who suggested that partial correlation is a better proxy to measure the connectivity among signals than correlation coefficient. (3) The RW-LASSO usually leads to the best sparsity, suggesting that the iterative re-weighting is an effective sparsity regularization approach. (4) Both SAR and SAR-harm have good performance even without supervising information. The comparison between the SAR and the SAR-noLC shows that introducing the link constraints as semi-supervision can further improve the accuracy of our approach.

Our approach has an important advantage regarding the computational efficiency by learning the shared regularization template, which is needed to construct the connectivity networks for new system instances. By learning a shared regularization template in the training process, our approach is not only more efficient in practice but also capable of visualizing the shared sparsity patterns.

7.2 Computational Efficiency

We evaluate the computational efficiency by simulating an application scenario with the PPMI data. First, we randomly select 500 patients for model training. Then, we choose five new patients arriving one by one for testing the models. We measure both the training time and the predicting time to support diagnosis for the new patients. Based on preliminary results, the number of features can significantly affect the running time. Thus, we evaluate the computational efficiency with different amounts of features. Table 5 summarizes the running time of our approach and the benchmarks, which are implemented with MATLAB 2017a, while FCNN is implemented with PyTorch. The experiments are conducted under the same computing environment: Ubuntu 16.04 LTS with Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz, 12 cores and 64G memory.

According to the results in Table 5, our SAR shows a significant advantage in terms of the computational efficiency. Specifically, in the training process, the SAR spends about 30% to 50% of the time required by MNR. When we set the number of features to 20, SAR takes 45.5% of the running time of MNR; setting the number of features to 200, SAR spends 162.27 seconds while MNR takes 426.79 seconds. Furthermore, our results demonstrate the strength of SAR in terms of the predicting time. For instance, SAR completes its predicting tasks with 200 features in 2.48 seconds, while MNR spends 2078.27 seconds to complete. WGraphSR is computationally expensive for both training and predicting due to the coupling of connectivity coefficients of correlated system signals. Overall, we find consistent advantage of our approach in terms of the computational efficiency.

7.3 Sparse Network Visualization

We can visualize the networks and the shared sparsity patterns to better understand important structures in the input data. Note that, although some of the benchmark methods can construct sparse networks, our approach is the only one that can identify shared sparsity patterns and further visualize the global connectivity structures. In the following, we show examples of such visualizations based on the PPMI data.

For highly sparse networks, we expect that non-zero values in the connectivity networks indicate important correlations between system signals. Thus, given a group of

TABLE 2: Comparisons of classification performance for **PPMI**

		Accuracy	Precision	Recall	$F1$	AUC	Sparsity
PCC	LR	0.710 ± 0.027	0.756 ± 0.019	0.852 ± 0.046	0.800 ± 0.021	0.755	39.9%
	SVM	0.688 ± 0.012	0.690 ± 0.009	0.990 ± 0.014	0.813 ± 0.007	0.612	
	FCNN	0.684 ± 0.013	0.686 ± 0.002	0.974 ± 0.032	0.805 ± 0.011	0.553	
PCC+Fisher	LR	0.644 ± 0.034	0.700 ± 0.017	0.838 ± 0.047	0.762 ± 0.027	0.596	39.9%
	SVM	0.683 ± 0.006	0.684 ± 0.002	0.997 ± 0.011	0.811 ± 0.005	0.524	
	FCNN	0.659 ± 0.024	0.696 ± 0.018	0.897 ± 0.087	0.781 ± 0.028	0.600	
SR-C	LR	0.781 ± 0.044	0.797 ± 0.041	0.916 ± 0.055	0.851 ± 0.031	0.797	41.8%
	SVM	0.687 ± 0.008	0.692 ± 0.008	0.976 ± 0.012	0.810 ± 0.004	0.778	
	FCNN	0.881 ± 0.019	0.922 ± 0.026	0.903 ± 0.012	0.912 ± 0.013	0.926	
PINV	LR	0.502 ± 0.044	0.674 ± 0.038	0.525 ± 0.050	0.590 ± 0.043	0.548	37.4%
	SVM	0.682 ± 0.003	0.683 ± 0.001	$\mathbf{0.998 \pm 0.004}$	0.811 ± 0.002	0.501	
	FCNN	0.657 ± 0.023	0.691 ± 0.008	0.903 ± 0.054	0.782 ± 0.021	0.591	
SR-PC	LR	0.865 ± 0.035	0.892 ± 0.031	0.914 ± 0.034	0.903 ± 0.025	0.935	83.2%
	SVM	0.830 ± 0.030	0.853 ± 0.021	0.910 ± 0.029	0.880 ± 0.020	0.914	
	FCNN	0.909 ± 0.013	0.934 ± 0.016	0.933 ± 0.029	0.933 ± 0.010	0.963	
RW-LASSO	LR	0.848 ± 0.029	0.881 ± 0.027	0.900 ± 0.033	0.890 ± 0.021	0.920	**96.6%**
	SVM	0.842 ± 0.024	0.883 ± 0.022	0.887 ± 0.032	0.885 ± 0.018	0.916	
	FCNN	0.806 ± 0.037	0.844 ± 0.040	0.882 ± 0.035	0.861 ± 0.026	0.853	
MNR	LR	0.914 ± 0.026	0.945 ± 0.020	0.929 ± 0.034	0.937 ± 0.020	0.972	94.3%
	SVM	0.905 ± 0.027	$\mathbf{0.936 \pm 0.021}$	0.924 ± 0.036	0.930 ± 0.021	**0.966**	
	FCNN	0.899 ± 0.026	0.909 ± 0.015	0.946 ± 0.028	0.927 ± 0.019	0.956	
WGraphSR	LR	0.756 ± 0.031	0.810 ± 0.027	0.842 ± 0.041	0.825 ± 0.024	0.819	93.6%
	SVM	0.747 ± 0.028	0.773 ± 0.023	0.894 ± 0.035	0.828 ± 0.019	0.800	
	FCNN	0.743 ± 0.052	0.794 ± 0.052	0.847 ± 0.026	0.819 ± 0.032	0.805	
SAR-noLC	LR	0.908 ± 0.023	0.918 ± 0.025	0.951 ± 0.023	0.934 ± 0.016	0.967	77.7%
	SVM	0.886 ± 0.023	0.917 ± 0.027	0.918 ± 0.030	0.917 ± 0.017	0.942	
	FCNN	0.922 ± 0.032	0.926 ± 0.043	$\mathbf{0.966 \pm 0.008}$	0.945 ± 0.021	0.975	
SAR	LR	0.908 ± 0.018	0.907 ± 0.018	$\mathbf{0.964 \pm 0.020}$	0.935 ± 0.012	0.972	84.6%
	SVM	0.891 ± 0.018	0.910 ± 0.022	0.934 ± 0.022	0.921 ± 0.013	0.954	
	FCNN	0.929 ± 0.013	0.929 ± 0.007	0.959 ± 0.017	0.944 ± 0.010	**0.978**	
SAR-harm	LR	$\mathbf{0.942 \pm 0.019}$	$\mathbf{0.962 \pm 0.018}$	0.953 ± 0.026	$\mathbf{0.957 \pm 0.014}$	**0.986**	84.1%
	SVM	$\mathbf{0.911 \pm 0.025}$	0.929 ± 0.031	0.942 ± 0.026	$\mathbf{0.935 \pm 0.018}$	0.961	
	FCNN	$\mathbf{0.940 \pm 0.018}$	$\mathbf{0.949 \pm 0.016}$	0.936 ± 0.025	$\mathbf{0.956 \pm 0.013}$	0.968	

TABLE 3: Comparisons of classification performance for **PS**

		Accuracy	Precision	Recall	$F1$	AUC	Sparsity
PCC	LR	0.625 ± 0.097	0.749 ± 0.091	0.707 ± 0.136	0.718 ± 0.081	0.635	5.0%
	SVM	0.665 ± 0.058	0.685 ± 0.024	0.951 ± 0.085	0.796 ± 0.044	0.626	
	FCNN	0.677 ± 0.075	0.737 ± 0.063	0.853 ± 0.143	0.780 ± 0.068	0.730	
PCC+Fisher	LR	0.649 ± 0.110	0.784 ± 0.124	0.711 ± 0.124	0.735 ± 0.081	0.695	5.0%
	SVM	0.677 ± 0.050	0.690 ± 0.023	$\mathbf{0.969 \pm 0.060}$	0.805 ± 0.035	0.622	
	FCNN	0.680 ± 0.091	0.751 ± 0.079	0.831 ± 0.151	0.777 ± 0.079	0.710	
SR-C	LR	0.729 ± 0.130	0.769 ± 0.083	0.867 ± 0.143	0.812 ± 0.103	0.752	86.6%
	SVM	0.673 ± 0.081	0.697 ± 0.040	0.933 ± 0.101	0.797 ± 0.059	0.705	
	FCNN	0.729 ± 0.073	0.746 ± 0.074	$\mathbf{0.942 \pm 0.056}$	0.829 ± 0.042	0.709	
PINV	LR	0.711 ± 0.132	0.786 ± 0.102	0.809 ± 0.156	0.790 ± 0.109	0.812	4.7%
	SVM	0.729 ± 0.098	0.761 ± 0.068	0.898 ± 0.115	0.820 ± 0.070	0.779	
	FCNN	0.735 ± 0.093	0.777 ± 0.092	0.889 ± 0.077	0.824 ± 0.058	0.824	
SR-PC	LR	0.766 ± 0.112	0.841 ± 0.097	0.827 ± 0.111	0.829 ± 0.082	0.871	86.9%
	SVM	$\mathbf{0.775 \pm 0.094}$	0.808 ± 0.064	0.893 ± 0.118	$\mathbf{0.844 \pm 0.071}$	0.854	
	FCNN	0.800 ± 0.090	0.860 ± 0.093	0.862 ± 0.065	0.858 ± 0.060	0.869	
RW-LASSO	LR	0.745 ± 0.103	0.763 ± 0.074	$\mathbf{0.924 \pm 0.095}$	0.833 ± 0.069	0.825	**92.3%**
	SVM	0.751 ± 0.114	$\mathbf{0.822 \pm 0.103}$	0.836 ± 0.129	0.820 ± 0.086	**0.875**	
	FCNN	0.797 ± 0.095	0.841 ± 0.103	0.893 ± 0.086	0.860 ± 0.062	0.871	
MNR	LR	0.763 ± 0.106	0.813 ± 0.100	0.871 ± 0.100	0.836 ± 0.073	0.840	89.8%
	SVM	0.705 ± 0.073	0.724 ± 0.049	0.933 ± 0.091	0.814 ± 0.049	0.785	
	FCNN	0.726 ± 0.107	0.783 ± 0.073	0.840 ± 0.122	0.807 ± 0.082	0.835	
WGraphSR	LR	0.757 ± 0.103	0.826 ± 0.084	0.836 ± 0.154	0.821 ± 0.087	0.846	74.3%
	SVM	0.695 ± 0.087	0.719 ± 0.063	0.929 ± 0.124	0.806 ± 0.067	0.786	
	FCNN	0.738 ± 0.087	0.786 ± 0.095	0.884 ± 0.120	0.823 ± 0.062	0.828	
SAR-noLC	LR	0.772 ± 0.082	0.800 ± 0.070	0.907 ± 0.100	0.845 ± 0.058	0.859	92.1%
	SVM	0.711 ± 0.078	0.736 ± 0.064	0.924 ± 0.105	0.814 ± 0.052	0.811	
	FCNN	0.815 ± 0.100	0.853 ± 0.092	0.898 ± 0.088	0.871 ± 0.069	**0.902**	
SAR	LR	$\mathbf{0.806 \pm 0.092}$	$\mathbf{0.856 \pm 0.077}$	0.880 ± 0.140	$\mathbf{0.859 \pm 0.077}$	**0.887**	82.8%
	SVM	0.748 ± 0.099	0.777 ± 0.070	0.902 ± 0.137	0.829 ± 0.077	0.820	
	FCNN	$\mathbf{0.825 \pm 0.112}$	$\mathbf{0.863 \pm 0.102}$	0.902 ± 0.091	$\mathbf{0.878 \pm 0.076}$	0.882	
SAR-harm	LR	0.794 ± 0.133	0.847 ± 0.099	0.862 ± 0.137	0.850 ± 0.103	0.861	85.9%
	SVM	0.695 ± 0.052	0.713 ± 0.034	0.942 ± 0.086	0.810 ± 0.039	0.771	
	FCNN	0.809 ± 0.107	0.841 ± 0.090	0.907 ± 0.112	0.867 ± 0.077	0.888	

TABLE 4: Comparisons of classification performance for **OD**

		Accuracy	Precision	Recall	$F1$	AUC	Sparsity
PCC	LR	0.663 ± 0.099	0.667 ± 0.106	$\mathbf{0.694 \pm 0.144}$	0.672 ± 0.096	0.686	
	SVM	0.638 ± 0.093	0.636 ± 0.094	0.687 ± 0.167	0.651 ± 0.103	0.675	14.7%
	FCNN	0.642 ± 0.095	0.641 ± 0.088	0.678 ± 0.143	0.652 ± 0.1003	0.683	
PCC+Fisher	LR	0.614 ± 0.093	0.609 ± 0.085	0.679 ± 0.169	0.632 ± 0.102	0.637	
	SVM	0.574 ± 0.091	0.560 ± 0.080	$\mathbf{0.857 \pm 0.146}$	0.668 ± 0.068	0.626	14.7%
	FCNN	0.511 ± 0.079	0.491 ± 0.293	0.486 ± 0.368	0.417 ± 0.258	0.523	
SR-C	LR	0.635 ± 0.060	$\mathbf{0.848 \pm 0.164}$	0.358 ± 0.107	0.489 ± 0.116	0.644	
	SVM	0.617 ± 0.064	0.800 ± 0.176	0.348 ± 0.099	0.472 ± 0.111	0.628	65.0%
	SVM	0.640 ± 0.085	0.833 ± 0.178	0.439 ± 0.191	0.533 ± 0.135	0.671	
PINV	LR	0.506 ± 0.074	0.513 ± 0.072	0.592 ± 0.135	0.542 ± 0.080	0.577	
	SVM	0.497 ± 0.060	0.407 ± 0.220	0.575 ± 0.381	0.456 ± 0.254	0.542	14.6%
	FCNN	0.551 ± 0.076	0.520 ± 0.131	0.664 ± 0.241	0.573 ± 0.161	0.594	
SR-PC	LR	0.710 ± 0.075	0.811 ± 0.111	0.568 ± 0.115	0.660 ± 0.095	0.741	
	SVM	0.661 ± 0.091	0.719 ± 0.138	0.579 ± 0.120	0.631 ± 0.094	0.693	79.4%
	FCNN	0.631 ± 0.123	0.699 ± 0.225	0.547 ± 0.208	0.584 ± 0.170	0.692	
RW-LASSO	LR	0.730 ± 0.085	0.846 ± 0.135	0.587 ± 0.115	0.685 ± 0.100	0.749	
	SVM	0.713 ± 0.102	0.805 ± 0.157	0.594 ± 0.132	0.673 ± 0.118	0.728	**88.1%**
	FCNN	0.689 ± 0.098	0.752 ± 0.134	0.598 ± 0.161	0.651 ± 0.127	0.718	
MNR	LR	0.722 ± 0.081	0.801 ± 0.123	0.614 ± 0.137	0.685 ± 0.103	0.795	
	SVM	0.730 ± 0.082	0.827 ± 0.130	0.598 ± 0.129	0.686 ± 0.111	0.714	48.0%
	FCNN	0.706 ± 0.139	0.791 ± 0.188	0.611 ± 0.153	0.676 ± 0.140	**0.793**	
WGraphSR	LR	0.666 ± 0.083	0.685 ± 0.101	0.648 ± 0.142	0.657 ± 0.094	0.765	
	SVM	0.647 ± 0.102	0.632 ± 0.097	0.720 ± 0.135	0.670 ± 0.103	0.742	47.8%
	FCNN	0.630 ± 0.081	0.791 ± 0.158	0.398 ± 0.149	0.507 ± 0.137	0.670	
SAR-noLC	LR	0.699 ± 0.085	0.785 ± 0.147	0.636 ± 0.189	0.673 ± 0.094	0.779	
	SVM	0.517 ± 0.014	0.250 ± 0.266	0.480 ± 0.510	0.329 ± 0.350	0.518	20.0%
	FCNN	0.732 ± 0.075	$\mathbf{0.853 \pm 0.124}$	0.593 ± 0.139;	0.684 ± 0.101	0.789	
SAR	LR	$\mathbf{0.734 \pm 0.068}$	0.821 ± 0.115	0.631 ± 0.118	$\mathbf{0.702 \pm 0.082}$	**0.815**	
	SVM	$\mathbf{0.746 \pm 0.066}$	$\mathbf{0.838 \pm 0.107}$	0.638 ± 0.118	$\mathbf{0.713 \pm 0.084}$	**0.753**	83.5%
	FCNN	0.723 ± 0.084	0.802 ± 0.107	0.621 ± 0.130	0.689 ± 0.094	0.787	
SAR-harm	LR	0.730 ± 0.072	0.823 ± 0.101	0.605 ± 0.142	0.687 ± 0.096	0.776	
	SVM	0.689 ± 0.076	0.734 ± 0.102	0.622 ± 0.139	0.663 ± 0.091	0.722	81.0%
	FCNN	$\mathbf{0.741 \pm 0.090}$	0.845 ± 0.122	0.608 ± 0.128	$\mathbf{0.698 \pm 0.112}$	0.792	

TABLE 5: Comparisons of running time (in seconds)

Stage	Method	Features									
		20	40	60	80	100	120	140	160	180	200
Predicting	PCC	0.00	0.00	0.01	0.01	0.01	0.01	0.02	0.02	0.02	0.03
	PCC+Fisher	0.02	0.01	0.02	0.01	0.03	0.04	0.02	0.04	0.05	0.06
	SR-C	0.56	0.91	1.57	2.75	3.96	7.55	10.02	11.73	15.75	18.70
	PINV	0.03	0.02	0.02	0.06	0.05	0.07	0.08	0.09	0.08	0.10
	SR-PC	1.51	1.52	1.54	1.56	1.64	1.64	1.77	1.77	2.01	1.83
	RW-LASSO	0.25	0.52	0.75	1.17	1.51	1.64	2.11	2.78	3.69	4.51
	SAR-noLC	0.08	0.23	0.35	0.49	0.59	0.80	0.83	0.85	1.25	2.27
	WGraphSR	0.24	3.89	20.68	65.26	157.12	319.53	602.48	1033.99	1670.00	3763.62
	MNR	34.96	110.05	208.64	331.16	541.52	762.71	1076.53	1373.08	1772.22	2078.27
	SAR	0.15	0.25	0.49	0.36	0.70	0.94	1.14	1.12	1.61	2.48
Training	PCC	0.35	0.55	1.28	1.80	3.58	6.31	8.90	13.16	17.17	18.45
	PCC+Fisher	0.57	0.74	1.64	1.99	4.65	7.77	10.58	13.71	18.60	23.00
	SR-C	3.07	11.33	19.57	33.70	50.29	109.33	146.78	215.24	222.99	248.01
	PINV	0.86	1.49	2.16	3.12	5.52	10.76	14.35	19.67	21.05	31.04
	SR-PC	14.77	10.72	13.27	16.47	21.64	27.51	33.53	38.52	47.09	51.69
	RW-LASSO	22.51	46.20	72.66	97.85	133.67	174.79	230.62	297.64	338.53	396.27
	SAR-noLC	1.94	4.36	9.08	15.84	32.66	44.11	61.08	73.58	99.66	120.44
	WGraphSR	34.20	408.90	2104.10	6392.20	15365.70	32381.80	60581.60	102704.50	165818.10	375280.10
	MNR	7.80	22.06	43.65	68.19	109.42	159.80	220.61	285.08	369.83	426.79
	SAR	3.55	6.95	12.76	21.33	43.43	56.08	83.05	100.66	137.93	162.27

connectivity networks, we compute the frequency of the non-zero values for each pair of nodes. For example, for all of the system instances from the same class, we define the *Connection Frequency Network* F as follows.

$$F(class) = \frac{\sum_{n \in class} sign\left(\left|\beta^{(n)}\right|\right)}{|\{n \in class\}|},$$

where $class$ represents the set of all the instances labeled with the same class type. In the PPMI data, we have two connection frequency networks, $F(\text{non-PD})$ and $F(\text{PD})$, for non-PD and PD subjects, respectively. The difference between the two networks, denoted by $DF = F(\text{non-PD}) - F(\text{PD})$, captures the difference between the two groups of connectivity networks constructed by our algorithm. Particularly, if $DF_{i,j} > 0$, features i and j will have a larger probability to be connected in the non-PD class, compared with the PD class. If $DF_{i,j} < 0$, we should expect a smaller connectivity probability between i and j in the non-PD class.

Figure 7 illustrates the top 100 values in matrix DF. We use red lines to represent the cases that the non-PD

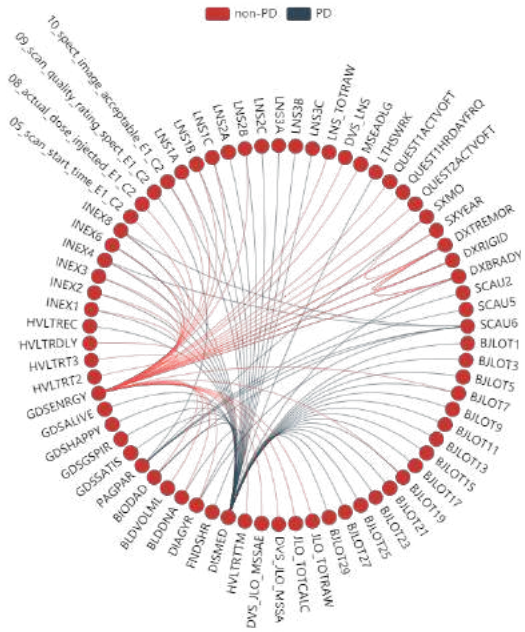


Fig. 7: The difference of connection frequency between the connectivity networks of non-PD and PD (PPMI).

has a higher probability of a strong connectivity, and the blue Lines are used for the cases that the PD has a larger probability. Let us focus on the following two features: GDSENRGY (Feel full of energy?) and DISMED (Use of PD Medication?). For non-PD, GDSENRGY will have a higher probability to be connected to other trials' results, such as LNS1A (Trial 1a), LNS1B (Trial 1b), and DVS_LNS (Derived-LNS Scaled Score). For PD, DISMED has a stronger influence on trial results, such as BJLOT (Benton Test Item), LNS1A, and LNS1B. These results can be further investigated by medicine research to support treatment decisions.

We can also visualize the shared regularization template for the global sparsity patterns. Figure 8 shows the top 100 connections from the template constructed by our approach based on the PPMI data. We find that CMDOSE (Dose of Concomitant Medications) has a strong relationship with some features in Geriatric Depression Scale, e.g., GDSEMPTY (Feel that your life is empty?) and GDSBORED (Often get bored?). CMDOSE is also strongly connected with Letter-Number Sequencing (PD), including LNS4C (Trial 4c) and LNS4B (Trial 4B). These connections in the network of the PD subjects are stronger than that of the non-PD subjects. Following that, further medicine studies may be conducted to confirm whether concomitant medications can relieve nervous problems for PD patients.

Finally, we compare the template identified by our approach and the pseudo-template by SR-PC [16]. As aforementioned, while our approach is capable of learning the shared template, the pseudo-template is computed with Equation 12 where the β matrices are optimized by SR-PC. As shown in Figure 9, the shared template in our approach is relatively sparser with a higher contrast ratio, indicating that our approach can differentiate strong and weak connections to construct informative networks with fewer edges (i.e., lower dimensions).

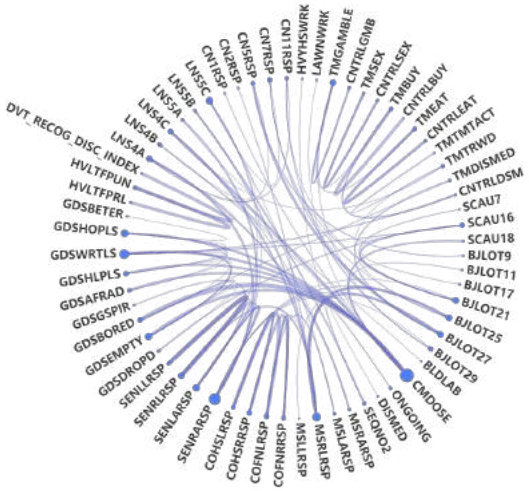


Fig. 8: The shared regularization template by our approach (PPMI).

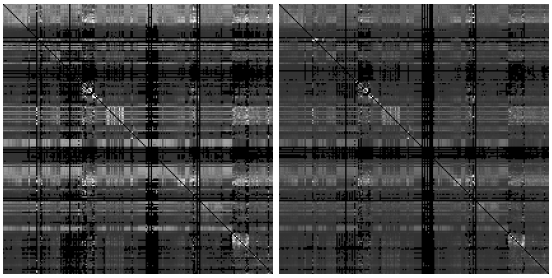


Fig. 9: The shared regularization template by our approach (left) and the pseudo-template by SR-PC (right).

8 Concluding Remarks

We developed a shared adaptive regularization (SAR) framework for inferring sparse connectivity networks from dynamic systems. In the network inference process, our approach can learn a regularization template shared by all the dynamic systems. Our approach can also incorporate supervising information so that the inferred networks lead to improved performance of predictive modeling tasks based on the connectivity networks. Overall, we developed a novel approach to modeling dynamic systems with shared adaptive regularization. The connectivity networks and the regularization template can be used in further learning tasks for the dynamic systems. To evaluate the effectiveness and the efficiency of our approach, our experiments used real-world data sets for several classification tasks. The results demonstrated the performance of our approach in comparison with several benchmarks.

Acknowledgments

Data used in the preparation of this article were obtained from the Parkinson's Progression Markers Initiative (PPMI) database (www.ppmi-info.org/data). For up-to-date information on the study, visit www.ppmi-info.org. PPMI – a public-private partnership – is funded by the Michael J. Fox Foundation for Parkinson's Research and funding partners,

including Abbvie, Allergan, Avid Radiopharmaceuticals, BioLegend, Bristol-Myers Squibb, Celgene, Denali, GE healthcare, Genentech, GlaxoSmithKline, Lilly, Lundbeck, Merck, Meso Scale Discovery, Pfizer, Piramal, Prevail therapeutics, Roche, SANOFI Genzyme, Servier, Takeda, Teva, UCB, Verily, Voyager, and Golub capital.

REFERENCES

[1] David Adametz and Volker Roth. Distance-based network recovery under feature correlation. In *Advances in Neural Information Processing Systems*, pages 775–783, 2014.

[2] Onureena Banerjee, Laurent El Ghaoui, Alexandre d'Aspremont, and Georges Natsoulis. Convex optimization techniques for fitting sparse gaussian graphical models. In *Proceedings of the 23rd international conference on Machine learning*, pages 89–96. ACM, 2006.

[3] Sugato Basu, Arindam Banerjee, and Raymond J Mooney. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the 2004 SIAM international conference on data mining*, pages 333–344. SIAM, 2004.

[4] Amir Beck and Marc Teboulle. Gradient-based algorithms with applications to signal recovery. *Convex optimization in signal processing and communications*, pages 42–88, 2009.

[5] Leo Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384, 1995.

[6] Luis M Candanedo and Véronique Feldheim. Accurate occupancy detection of an office room from light, temperature, humidity and co2 measurements using statistical learning models. *Energy and Buildings*, 112: 28–39, 2016.

[7] Emmanuel J Candès and Michael B Wakin. An introduction to compressive sampling. *IEEE signal processing magazine*, 25(2):21–30, 2008.

[8] Emmanuel J Candes, Michael B Wakin, and Stephen P Boyd. Enhancing sparsity by reweighted l1 minimization. *Journal of Fourier analysis and applications*, 14(5-6): 877–905, 2008.

[9] Lei Du, Heng Huang, Jingwen Yan, Sungeun Kim, Shannon L Risacher, Mark Inlow, Jason H Moore, Andrew J Saykin, Li Shen, and Alzheimer's Disease Neuroimaging Initiative. Structured sparse canonical correlation analysis for brain imaging genetics: an improved graphnet method. *Bioinformatics*, 32(10):1544–1551, 2016.

[10] Hongyang Gao, Zhengyang Wang, and Shuiwang Ji. Large-scale learnable graph convolutional networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1416–1424. ACM, 2018.

[11] Kwang-Il Goh, Michael E Cusick, David Valle, Barton Childs, Marc Vidal, and Albert-László Barabási. The human disease network. *Proceedings of the National Academy of Sciences*, 104(21):8685–8690, 2007.

[12] Heng Huang, Jingwen Yan, Feiping Nie, Jin Huang, Weidong Cai, Andrew J Saykin, and Li Shen. A new sparse simplex model for brain anatomical and genetic network analysis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 625–632. Springer, 2013.

[13] Shuai Huang, Jing Li, Liang Sun, Jun Liu, Teresa Wu, Kewei Chen, Adam Fleisher, Eric Reiman, and Jieping Ye. Learning brain connectivity of alzheimer's disease from neuroimaging data. In *NIPS*, volume 22, pages 808–816, 2009.

[14] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

[15] Brian Kulis, Sugato Basu, Inderjit Dhillon, and Raymond Mooney. Semi-supervised graph clustering: a kernel approach. *Machine learning*, 74(1):1–22, 2009.

[16] Hyekyoung Lee, Dong Soo Lee, Hyejin Kang, Boong-Nyun Kim, and Moo K Chung. Sparse brain network recovery under compressed sensing. *Medical Imaging, IEEE Transactions on*, 30(5):1154–1165, 2011.

[17] Chenlei Leng, Yi Lin, and Grace Wahba. A note on the lasso and related procedures in model selection. *Statistica Sinica*, pages 1273–1284, 2006.

[18] Jason Lines and Anthony Bagnall. Time series classification with ensembles of elastic distance measures. *Data Mining and Knowledge Discovery*, 29(3):565–592, 2015.

[19] Nicolai Meinshausen and Peter Buhlmann. High-dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34(3):1436–1462, 2006.

[20] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

[21] Jie Peng, Pei Wang, Nengfeng Zhou, and Ji Zhu. Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, 104 (486), 2009.

[22] Betul Erdogdu Sakar, M Erdem Isenkul, C Okan Sakar, Ahmet Sertbas, Fikret Gurgen, Sakir Delil, Hulya Apaydin, and Olcay Kursun. Collection and analysis of a parkinson speech dataset with multiple types of sound recordings. *IEEE Journal of Biomedical and Health Informatics*, 17(4):828–834, 2013.

[23] Juliane Schäfer and Korbinian Strimmer. An empirical bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21(6):754–764, 2004.

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

[25] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018. URL http://arxiv.org/abs/1710.10903.

[26] Hua Wang, Heng Huang, and Chris Ding. Correlated protein function prediction via maximization of data-knowledge consistency. *Journal of Computational Biology*, 22(6):546–562, 2015.

[27] Kun Wang, Meng Liang, Liang Wang, Lixia Tian, Xinqing Zhang, Kuncheng Li, and Tianzi Jiang. Altered functional connectivity in early alzheimer's disease: A resting-state fmri study. *Human brain mapping*, 28(10):

967–978, 2007.

[28] Chong-Yaw Wee, Pew-Thian Yap, Daoqiang Zhang, Lihong Wang, and Dinggang Shen. Constrained sparse functional connectivity networks for mci classification. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2012*, pages 212–219. Springer, 2012.

[29] Shiming Xiang, Feiping Nie, and Changshui Zhang. Learning a mahalanobis distance metric for data clustering and classification. *Pattern recognition*, 41(12): 3600–3612, 2008.

[30] Renping Yu, Lishan Qiao, Mingming Chen, Seong-Whan Lee, Xuan Fei, and Dinggang Shen. Weighted graph regularized sparse brain network construction for mci identification. *Pattern recognition*, 90:220–231, 2019.

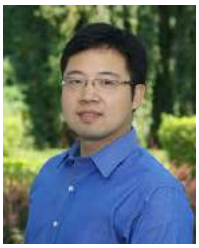
[31] Jie Zhang, Wei Cheng, ZhengGe Wang, ZhiQiang Zhang, WenLian Lu, GuangMing Lu, and Jianfeng Feng. Pattern classification of large-scale functional brain networks: identification of informative neuroimaging markers for epilepsy. *PloS one*, 7(5):e36733, 2012.

[32] Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476): 1418–1429, 2006.

Yue Qu is pursuing his PhD degree in Computer Science at the Dalian University of Technology. He received his B.E. degree in biomedical engineering and M.S. degree in computer science and technology from the Dalian University of Technology, Dalian, China, in 2013 and 2020, respectively. His research interests include data mining, information systems, and intelligent computing.

Chuanren Liu received the BS degree from the University of Science and Technology of China (USTC), the MS degree from the Behang University (BUAA), and the PhD degree from Rutgers, the State University of New Jersey. His research interests include data mining and knowledge discovery, and their real-world applications in business analytics.

Kai Zhang received his Master's degree from the Institute of Automation, Chinese Academy of Sciences in 2004, and his PhD degree in Computer Science in 2008 from the Hong Kong University of Science and Technology. Kai has devoted his research to machine learning, bioinformatics, brain functional networks and time series/complex network modelling.



Keli Xiao is an associate professor with the College of Business, Stony Brook University. He received his PhD degree in management from Rutgers, the State University of New Jersey in 2013. His research interests include business analytics, data analytics, and quantitative finance.

Bo Jin is a full professor at Dalian University of Technology. He received the Ph.D. degree from the Dalian University of Technology. He is an Associate Editor of IEEE ACCESS. He has served regularly on the organization and program committees of numerous conferences, including KDD, ICDM, AAAI and IJCAI.

Hui Xiong is a distinguished professor at Rutgers, the State University of New Jersey. He received the Ph.D. degree from the University of Minnesota (UMN), USA. His general area of research is data and knowledge engineering with a focus on developing effective and efficient techniques for emerging data intensive applications. He is an IEEE Fellow and an ACM Distinguished Scientist.