Subnetwork Mining with Spatial and Temporal Smoothness

Xuan Hong Dang, Hongyuan You, Ambuj K. Singh* Scott Grafton†

Abstract

In many real-world applications, data is represented in the form of networks with structures and attributes changing over time. The dynamic changes not only happen at nodes/edges, forming local subnetwork processes, but also eventually influence global states of networks. The need to understand what these local network processes are, how they evolve and consequently govern the progression of global network states has become increasingly important. In this paper, we explore these questions and develop a novel algorithm for mining a succinct set of subnetworks that are predictive and evolve along with the progression of global network states. Our algorithm is designed in the framework of logistic regression that fits a model for multi-states of network samples. Its objective function considers both the spatial network topology and temporal smooth transition between adjacent global network states, and we show that its global optimum solution can be achieved via steepest descent. Extensive experimental analysis on both synthetic and real world datasets demonstrates the effectiveness of our algorithm against competing methods, not only in the prediction accuracy but also in terms of domain relevance of the discovered subnetworks.

1 Introduction

Network data arises in a number of application domains ranging from neuroscience, biology, geography, to social sciences [2,21]. Accordingly, network analysis has emerged as a major paradigm for exploring the complex processes behind the observed network data. Compared to high dimensional data, the analysis over the network data is more challenging due to the nature of inter-dependence among the data entities. Moreover, most network data are not static but keep changing over time. These changes not only happen at the local interactions within the network samples but also eventually influence the global behaviors of the network instances. Consider brain networks associated with the Alzheimer's disease (AD) as an example [1]. The death of neurons and synapses can impact the cognitive performance of some brain regions. At an early stage, the disease can be as mild as making the patient difficult in remembering some recent events; however, as the disease progresses with more brain regions being impacted due to the neuronal connectivity, the decline of overall cognitive function can be experienced, including confusion, difficulty speaking, and eventually fatality. Detecting brain subnetwork markers that highly predict and evolve along with the progression of the disease is thus an important task since it not only helps to characterize the disease but further provides the means to plan the right treatments at the right stage of the disease.

Analyzing network structural data has been widely studied in the literature with existing work focusing on community extraction [21], frequent subgraph mining [10], outlier detection [2], and graph classification [12]. Although these studies have substantially promoted our understanding, they tend to be explored in a simple setting of a single network (e.g., community/cluster discovery), or extended to multiple-network settings (e.g., frequent subgraph mining), yet without fully investigating the essential relationship between local network processes and global network behaviors. In this paper, we broaden the research scope to a more complex setting in which we deal with multiple network samples, each of which is associated with a global label that reflects the current state of the entire network (e.g., disease stage, climate condition). Both local network interaction and global network state can evolve over time. The task is to uncover a small set of local network processes that have maximum impact on the global network states such that their evolution can be used to predict the evolution of global network states.

Certainly, one may argue that it is possible to view each network sample as a collection of edges and apply typical feature selection techniques like mutual information [20], statistical t-test [9], or perform PCA/SVD to generate new features prior to the analysis. While such an approach significantly reduces the number of features to be analyzed, the obtained models often lack domain relevance since the nature of interaction among network entities is completely ignored. A more feasible approach is to apply an effective graph classification method [12] to categorize network samples with different global states. Though this approach can be feasible for the goal of prediction, its newly generated features (typically in the form of frequent subgraphs [4,12]) lack temporal smoothness, and thus are less successful in

*Department of Computer Science, UCSB
†Department of Psychological and Brain Sciences, UCSB

interpreting and explaining the intrinsic network processes governing global network properties.

We present a novel algorithm to discover a succinct set of informative subnetworks that are highly predictive w.r.t. the development of the global network states. Due to the smooth transition of global states, these local subnetwork processes do not change abruptly from state to state, but rather develop smoothly along with the progression of the network states. Our algorithm is developed in the framework of logistic regression that fits the model for multi-states of network samples. Each global state of networks is characterized by a parametric vector whose coefficients are learnt with regularization on both the network topology (i.e., spatial smoothness) and the transition between any two adjacent global network states (i.e., temporal smoothness). In order to ensure that only the most predictive subnetworks will be learnt, we further exploit the sparsity-inducing $L1$-norm imposed on each parametric vector to remove edges that have little or no impact on the progression of global network states. The proposed formulation is challenging to solve due to the introduction of non-smooth $L1$-norm term. We, however, will show that the developed function is convex and our solution based on the steepest descent is practically efficient. Our contribution can be summarized as follows: (i) We propose and motivate a novel problem to study the impact of local subnetwork processes on global network properties that both evolve over time. (ii) We propose an objective function to learn local subnetwork processes that are highly predictive for the development of global network states. (iii) We theoretically prove that the formulated function is convex (though not strictly) and develop a novel gradient descent algorithm to optimize for a global optimum solution. (iv) We extensively evaluate and demonstrate the appealing performance of the proposed technique on both synthetic and real-world applications.

2 Preliminaries

DEFINITION 2.1. (Network sample) *Let $S^{(i)} = (\mathcal{N}, \mathcal{E}_i)$ be a network sample, where $\mathcal{N}$ is a set of pre-defined nodes, $\mathcal{E}_i \subseteq \mathcal{N} \times \mathcal{N}$ is a set of undirected edges. There is a labeling function $\mathcal{F}$ operating on local edges' values that maps each edge to a real number $\mathcal{F} : \mathcal{E}_{uv}^{(i)} \to \mathbb{R}$.*

We denote $\mathcal{DS} = \{S^{(1)}, S^{(2)}, \ldots, S^{(n)}\}$ as the database that consists of n network samples. Each network $S^{(i)}$ is further associated with a label y_i, indicating its global network state. Each y_i receives a discrete value in $\{1, 2, \ldots, K\}$ and this order reflects the temporal evolution over global network states. It is important to note that indexing (i)'s are *not* networks' timestamps. Instead, we consider temporal development based on values of global network states. For example, $S^{(i)}$'s can be snapshots captured from a social or traffic network in *multiple* days, and y_i's reflect whether such snapshots are in the morning, afternoon, or evening hours. Likewise, $S^{(i)}$'s can be brain networks scanned from *multiple* subjects, and associated y_i's label their disease-stages advanced from mild to moderate to severe condition. The temporal development is thus at the *group* level, instead of a network individual. For each global state, we define an aggregate graph that generalizes network topology of all network samples having that network state. Prior to that, let us define the adjacency of two edges in a network sample as follows.

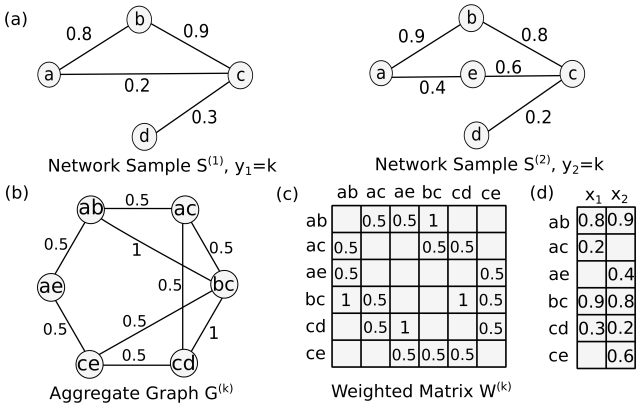

Fig. 1: (a) network samples; (b) aggregate graph; (c) weighted matrix; (d) vectors encoding edge values.

DEFINITION 2.2. (Edge Adjacency) *Let $S^{(i)} = (\mathcal{N}, \mathcal{E}_i)$ be a network sample characterized by Def.2.1, we define a pair $\{\mathcal{E}_{uv}^{(i)}, \mathcal{E}_{st}^{(i)}\} \in \mathcal{E}^{(i)}$ as adjacent edges if they have one node in common; otherwise, we call $\{\mathcal{E}_{uv}^{(i)}, \mathcal{E}_{st}^{(i)}\}$ non-adjacent or distant edges of $S^{(i)}$.*

DEFINITION 2.3. (Aggregate Graph for k-th state) *Let $\mathcal{DS}_k = \{S^{(i)} \in \mathcal{DS} | y_i = k\}$, we define $\mathcal{G}^{(k)} = (\mathcal{V}^{(k)}, \mathcal{L}^{(k)}, \mathcal{W}^{(k)})$ as an aggregate graph that captures edge adjacency of all networks in $\mathcal{DS}_k$, where $\mathcal{V}^{(k)} = \{v_1, \ldots, v_m\}$ is the vertex set with each v_p corresponds to an edge found in at least one $S^{(i)} \in \mathcal{DS}_k$; $\mathcal{L}^{(k)} \subseteq \mathcal{V}^{(k)} \times \mathcal{V}^{(k)}$ is the set of links. A link $\mathcal{L}_{pq}^{(k)}$ connects vertices v_p and v_q if the two corresponding edges are adjacent in at least one $S^{(i)} \in \mathcal{DS}_k$ according to Def.2.2. Each link $\mathcal{L}_{pq}^{(k)}$ is associated with a non-negative value $\mathcal{W}_{pq}^{(k)}$ encoding the fraction of $S^{(i)}$'s in $\mathcal{DS}_k$ having that edge-adjacency in their network structures.*

Fig.1 shows a simple example to illustrate the concepts presented in the three definitions above. Two network samples having the same k-th state are shown in Fig.1(a) while their aggregate graph $\mathcal{G}^{(k)}$ is depicted in Fig.1(b). A value of 0.8 associated with edge $\{ab\}$ in $S^{(1)}$ encodes the degree to which nodes a and b are related in $S^{(1)}$, whilst the value of 1 associated with link $\{ab, bc\}$ in $\mathcal{G}^{(k)}$ indicates that two edges $\{ab\}$ and

$\{bc\}$ are found adjacent in both $S^{(1)}$ and $S^{(2)}$. In the following, we refer to aggregate graphs shortly as *graphs*, and associate the terms "vertex" and "link" specifically with such graphs. Fig.1(c) shows $\mathcal{W}^{(k)}$ as a matrix representing graph $\mathcal{G}^{(k)}$.

3 Multinomial Logistic Regression

For the class of network applications addressed in this study, though network samples may have considerably different values associated with local edges, their network topologies are generally stable across samples (e.g., human brain networks, snapshots from a sensor or social network, etc.). We therefore utilize a high dimensional vector $x_i = [x_{i1}, \ldots, x_{im}]^T \in \mathcal{R}^m$ to store the local edge values of a network sample $S^{(i)}$ (examples of x_i's are illustrated in Fig.1(d)). Under the framework of logistic regression, we directly model the posterior probability of a k-th state imposed on an input sample x_i as follows:

$$P(y_i = k|x_i, \Theta, b) = \frac{\exp\left(\theta_k^T x_i - b_k\right)}{\sum_{j=1}^{K} \exp\left(\theta_j^T x_i - b_j\right)} \tag{3.1}$$

where b_k and θ_k are respectively the bias term and the weight vector that characterize the corresponding class k. Across all K models, scalars b_k's form the bias vector b while vectors θ_k's form the matrix Θ. They both are the parameters to be estimated and with the maximum likelihood optimization, their coefficients can be fitted by minimizing the negative log likelihood over n network samples:

$$NLL(\Theta, b) = -\log \prod_{i=1}^{n} P(y_i = k|x_i, \Theta, b) \tag{3.2}$$

4 Spatial and Temporal Smoothness

Spatial Smoothness: Network topology plays a key role in determining how nodes communicate. In order to discover subnetworks related to global network states, the connectivity patterns within network samples have to be taken into account. Toward this goal, within each k-th class of network samples, we aim to regularize the parametric vector θ_k subject to the topology captured by $\mathcal{G}^{(k)}$, through using its Laplacian matrix $C^{(k)}$ defined by:

$$C_{pq}^{(k)} = \begin{cases} 1 - \mathcal{W}_{pq}^{(k)}/d_p & \text{if } v_p = v_q \text{ and } d_p \neq 0 \\ -\mathcal{W}_{pq}^{(k)}/\sqrt{d_p d_q} & \text{if } v_p \text{ and } v_q \text{ are connected in } \mathcal{G}^{(k)} \\ 0 & \text{otherwise} \end{cases}$$

in which d_p is the vertex degree of v_p. Thus, the negative log likelihood is incorporated with the spatial network constraint term as follows:

$$F(\Theta, b) = -\log \prod_{i=1}^{n} P(y_i{=}k|x_i, \Theta, b) + \frac{\lambda_1}{2} \sum_k \theta_k^T C^{(k)} \theta_k \tag{4.3}$$

with $\lambda_1 \geq 0$, and we can further expand:

$$\theta_k^T C^{(k)} \theta_k = \sum_{v_p} \sum_{v_q} \left(\frac{\theta_k(p)}{\sqrt{d_p}} - \frac{\theta_k(q)}{\sqrt{d_q}} \right)^2 \mathcal{W}_{pq}^{(k)} \tag{4.4}$$

From this equation, it is clearly seen that if $\mathcal{W}_{pq}^{(k)}$ is large, indicating two vertices v_p and v_q in graph $\mathcal{G}^{(k)}$ strongly interact in a large portion of the network samples, the coefficients at p-th and q-th entries of vector θ_k should be similar, i.e., smooth, in order to minimize this equation. That means the selection of either v_p or v_q will encourage the selection of the other one due to the large $\mathcal{W}_{pq}^{(k)}$, leading to the formation of subnetworks selected in the final results. This network term is thus considered as the spatial smoothness since it minimizes the difference between connected vertices in the aggregate graph.

Temporal Smoothness: Given the smooth progression on the observable global network states, it is particularly important to capture the temporal changes in the network connectivity patterns developed along with this process. For any two consecutive global states, one would expect that the local subnetwork processes influencing the development of global network states will evolve in a smooth way rather than in an abrupt manner. Following our objective function developed above, we thus further develop another regularization term that penalizes large deviations between any two parametric vectors θ_k and θ_{k+1}, which respectively account for the two consecutive global network states. This gives rise to the following formulation:

$$\begin{aligned} F(\Theta, b) = &-\log \prod_{i=1}^{n} P(y_i = k|x_i, \Theta, b) + \frac{\lambda_1}{2} \sum_k \theta_k^T C^{(k)} \theta_k \\ &+ \frac{\lambda_2}{2} \sum_{k=1}^{K-1} \|\theta_{k+1} - \theta_k\|_2^2 \end{aligned} \tag{4.5}$$

where $\lambda_2 \geq 0$ is a parameter regularizing the impact of temporal smoothness. It is worth noting that the L_2 norm on the parametric vectors encourages coefficients at a p-th feature across different states to be grouped together, which is essential for tracking the temporal changes of various local network processes along with the evolution of the global network states. Moreover, though each θ_k is learnt for a global state and their coefficients can be different in scale, such parametric vectors are usually very sparse with majority of entries are zeros (presented in the next section) in order to uncover a small set of predictive subnetworks. The temporal smoothness thus can be interpreted as penalizing the dissimilarity in sparseness across network samples with adjacent global network states, ensuring the smooth changes in the succinct set of uncovered predictive subnetworks.

5 Optimization with sparse model

In solving the objective optimization function formulated in Eq.(4.5), we resort to the steepest gradient descent method since there is no closed form solution for Eq.(4.5). For convenience, network state

y_i is re-formatted as "1-of-K" encoding vector $\mathbf{y}_i = [\mathbf{y}_i^1, \ldots, \mathbf{y}_i^K]^T$ such that only $\mathbf{y}_i^k = 1$ and all other entries are 0 if the original scalar $y_i = k$ (e.g., if $K = 3$ and $y_i = 2$, then $\mathbf{y}_i = [0, 1, 0]$). Following this, the conditional likelihood of y_i given x_i can be written as:

$$P(y_i = k|x_i, \Theta, b) = \prod_{k=1}^{K} P(\mathbf{y}_i^k|x_i, \Theta, b)^{\mathbf{y}_i^k} \tag{5.6}$$

For brevity, we denote $P(k|x_i)^{\mathbf{y}_i^k}$ for $P(\mathbf{y}_i^k|x_i, \Theta, b)^{\mathbf{y}_i^k}$ by omitting parameter set $\{\Theta, b\}$. And its log form can be expanded by: $\log P(k|x_i)^{\mathbf{y}_i^k} = \sum_k \mathbf{y}_i^k(\theta_k^T x_i + b_k) - \log \sum_j exp(\theta_j^T x_i + b_j)$.

In this form, it is straightforward to show that function $F(\Theta, b)$ in Eq.(4.5) is strictly convex and the optimal solution for parameters θ_k's and b_k's can be achieved through iterative updates with their first derivatives given by:

$$\begin{aligned} F'(\theta_k) &= \frac{\partial F}{\partial \theta_k} = \sum_i (P(k|x_i) - \mathbf{y}_i^k)x_i + \lambda_1 C^{(k)}\theta_k \\ &\quad - \lambda_2(\theta_{k-1} - \theta_k) + \lambda_2(\theta_k - \theta_{k+1}) \quad \text{for } \theta_k's \end{aligned} \tag{5.7}$$

$$F'(b_k) = \frac{\partial}{\partial b_k} = \sum_i (P(k|x_i) - \mathbf{y}_i^k) \quad \text{for } b_k's. \tag{5.8}$$

However, such a solution does not set any coefficient of θ_k's to zero and thus, no predictive subnetworks are selected which lacks crucial point of interpretation. We extend our solution to the more challenging optimization by further imposing the sparseness on θ_k's through constraining their L1-norm within a constant value $t > 0$:

$$\begin{aligned} f(\Theta, b) = &- \log \prod_{i=1}^{n}\prod_{k=1}^{K} P(k|x_i)^{\mathbf{y}_i^k} + \frac{\lambda_1}{2}\sum_k \theta_k^T C^{(k)}\theta_k \\ &+ \frac{\lambda_2}{2}\sum_{k=1}^{K-1} \|\theta_{k+1} - \theta_k\|_2^2 \qquad \text{subject to } \sum_k |\theta_k| \le t \end{aligned} \tag{5.9}$$

The advantage of L1-norm constraint [7] is that it causes many coefficients to be exactly zero when t is set sufficiently small. Together with our network regularization terms, this constraint naturally performs subnetwork discovery by keeping only the most relevant substructures that are predictive to global network states. Eq.(5.9) can be equivalently formulated in the Lagrangian form as follows:

$$\begin{aligned} f(\Theta, b) = &- \log \prod_{i=1}^{n}\prod_{k=1}^{K} P(k|x_i)^{\mathbf{y}_i^k} + \frac{\lambda_1}{2}\sum_k \theta_k^T C^{(k)}\theta_k \\ &+ \frac{\lambda_2}{2}\sum_{k=1}^{K-1} \|\theta_{k+1} - \theta_k\|_2^2 + \beta\sum_{k=1}^{K} |\theta_k| \end{aligned} \tag{5.10}$$

with $\beta > 0$ playing the role of t, but a larger value of β forces more coefficients in θ_k's equal to 0. Unlike linear regression problem, our optimization function $f(\Theta, b)$ here is no longer strictly convex and obviously, there exists no closed form expression for it. Before showing Eq.(5.10) is minimizable via the approach of steepest descent, let us denote $F(\theta_k)$ for a single network state by:

$$\begin{aligned} F(\theta_k) = &- \log \prod_i P(k|x_i)^{\mathbf{y}_i^k}\theta_k + \frac{\lambda_1}{2}\theta_k^T C^{(k)}\theta_k \\ &+ \frac{\lambda_2}{2}\left(\|\theta_{k+1} - \theta_k\|_2^2 + \|\theta_k - \theta_{k-1}\|_2^2\right) \end{aligned} \tag{5.11}$$

For simplification, we have included the bias term b_k as the first entry of the parametric vector θ_k and thus it is skipped in the notation of $F(\theta_k)$. Note that this bias coefficient is *excluded* from both the network regularization and the model smoothness terms. In optimizing each single parametric vector θ_k, Eq.(5.10) can be simply written as:

$$f(\theta_k) = F(\theta_k) + \beta|\theta_k| \tag{5.12}$$

We have the following theorem:

THEOREM 1. *Given the spatial smoothness and temporal smoothness defined in Eq.*(4.3)*&*(4.5)*, function* $F(\theta_k)$ *defined Eq.*(5.11) *is convex.*

Proof: The proof of this theorem is straightforward by showing that its Hessian matrix is positive definite. More specifically, continuing with the first derivative shown in Eq.(5.7), it can be shown that the second derivative of $F(\theta_k)$ with respect to θ_k is:

$$\begin{aligned} F''(\theta_k) &= \sum_i x_i x_i^T (P(k|x_i) - P(k|x_i)^2) + \lambda_1 C^{(k)} + \lambda_2 I \\ &= X P^{(k)} X^T + \lambda_1 \left(C^{(k)} + \frac{\lambda_2}{\lambda_1} I\right) \end{aligned} \tag{5.13}$$

where $P^{(k)}$ is the diagonal matrix with entries $(P(k|x_i) - P(k|x_i)^2)$'s, X is the matrix with x_i's as its columns and I is the identity matrix. It is obvious that, given any non-zero vector $u \in \mathcal{R}^m$, the quadratic form $u^T F''(\theta_k) u > 0$ since: (i) $(P(k|x_i) - P(k|x_i)^2) \ge 0$ as $0 \le P(k|x_i) \le 1$; (ii) $C^{(k)}$ is positive definite according to Eq.(4.4); (iii) and both λ_1 and λ_2 are non-negative numbers by our setting. □

THEOREM 2. *Given* $f(\theta_k) = F(\theta_k) + \beta|\theta_k|$ *where* β *is non-negative, our optimization function* $f(\theta_k)$ *is convex.*

Proof: Let $h(\theta_k) = \beta|\theta_k|$ and for any $\theta_k^{(1)}, \theta_k^{(2)}$ defined in a convex domain, $\zeta \in (0, 1)$, and $\theta_k = \zeta\theta_k^{(1)} + (1-\zeta)\theta_k^{(2)}$, then $h(\theta_k)$ is also a convex function since:

$$\begin{aligned} &\zeta h(\theta_k^{(1)}) + (1-\zeta)h(\theta_k^{(2)}) = \beta\zeta|\theta_k^{(1)}| + \beta(1-\zeta)|\theta_k^{(2)}| \\ &= \beta|\zeta\theta_k^{(1)}| + \beta|(1-\zeta)\theta_k^{(2)}| \ge \beta|\zeta\theta_k^{(1)} + (1-\zeta)\theta_k^{(2)}| = h(\theta_k) \end{aligned}$$

in which the triangle inequality has been used in the second row and given $\beta \ge 0$. In combination with Theorem 1, and note that the summation of convex functions defined in the convex domain is also convex [3], it follows that $f(\theta_k)$ is a convex function. □

Theorem 2 is important as it ensures that our steepest descent algorithm will converge. In the spirit of gradient descent, our algorithm keeps updating parametric

vector θ_k by $\theta_k + \delta_k$ as long as the overall function is being reduced. Let us specify:

$$f(\theta_k) = F(\theta_k) + \beta|\theta_k| \quad \text{and} \tag{5.14}$$

$$f(\theta_k + \delta_k) = F(\theta_k + \delta_k) + \beta|\theta_k + \delta_k| \tag{5.15}$$

Then the algorithm finds δ_k such that $f(\theta_k) - f(\theta_k + \delta_k)$ is maximized. That means the next step of moving θ_k will lead to the steepest descent in reducing $f(\theta_k)$. This maximization is equivalent to minimization of $f(\theta_k + \delta_k) - f(\theta_k)$:

$$\begin{aligned} \underset{\delta_k}{\text{argmin}}\ g(\delta_k) &= f(\theta_k + \delta_k) - f(\theta_k) \\ &= F(\theta_k + \delta_k) - F(\theta_k) + \beta(|\theta_k + \delta_k| - |\theta_k|) \end{aligned} \tag{5.16}$$

Further formulating terms on the right hand size, the 2nd order Taylor expansion can be exploited to get:

$$F(\theta_k + \delta_k) = F(\theta_k) + \delta_k^T F'(\theta_k) + \frac{1}{2}\delta_k^T F''(\theta_k)\delta_k \text{ or}$$

$$F(\theta_k + \delta_k) - F(\theta_k) = \delta_k^T F'(\theta_k) + \frac{1}{2}\delta_k^T F''(\theta_k)\delta_k \tag{5.17}$$

where $F'(\theta_k)$ and $F''(\theta_k)$ are respectively the gradient vector (Eq.(5.7)), and the Hessian (Eq.(5.13)). Then, Eq.(5.16) can be rewritten as:

$$\underset{\delta_k}{\text{argmin}}\, g(\delta_k) = F'(\theta_k)^T\delta_k + \frac{1}{2}\delta_k^T F''(\theta_k)\delta_k + \beta\left(|\theta_k+\delta_k|-|\theta_k|\right)$$

However, optimizing this function is still challenging since it is not smooth due to the two absolute terms. Therefore, in order for minimization, the subderivative of the function is required. For simplicity, we provide the calculation w.r.t. each coefficient of δ_k. In this case, let δ_{kj} be the j-th entry of δ_k, then the function can be written by:

$$g(\delta_{kj}) = F'_j(\theta_k)\times\delta_{kj} + \frac{1}{2}F''_{jj}(\theta_k)\times\delta_{kj}^2 + \beta(|\theta_{kj} + \delta_{kj}|-|\theta_{kj}|)$$

where $F'_j(\theta_k)$ is the j-th entry of the gradient vector $F'(\theta_k)$, and $F''_{jj}(\theta_k)$ is the j-th entry in the diagonal of the Hessian matrix $F''(\theta_k)$. Consequently, we separate the subderivative w.r.t. δ_{kj} into the following cases:

$$\frac{\partial g(\delta_{kj})}{\partial\delta_{kj}} = \begin{cases} F'_j(\theta_k) + F''_{jj}(\theta_k)\delta_{kj} + \beta & \text{if } \delta_{kj} > -\theta_{kj} \\ F'_j(\theta_k) + F''_{jj}(\theta_k)\delta_{kj} - \beta & \text{if } \delta_{kj} < -\theta_{kj} \end{cases}$$

Note that the derivative of the absolute term is not defined when $\theta_{kj} - \delta_{kj} = 0$, or in this case we have $\delta_{kj} = -\theta_{kj}$.

Now we know that δ_{kj} is optimal if the minimum-norm subgradient at δ_{kj} is equal to zero. Thus, combining with the setting of the subderiviate to zero yields:

$$\delta_{kj} = \begin{cases} -\frac{F'_j(\theta_k)+\beta}{F''_{jj}(\theta_k)} & \text{if } F'_j(\theta_k) < F''_{jj}(\theta_k)\theta_{kj} - \beta \\ -\frac{F'_j(\theta_k)-\beta}{F''_{jj}(\theta_k)} & \text{if } F'_j(\theta_k) > F''_{jj}(\theta_k)\theta_{kj} + \beta \\ -\theta_{kj} & \text{otherwise} \end{cases} \tag{5.18}$$

For the bias term b_k, which we previously included as the top entry of θ_k, its corresponding deviation is $\delta_{k0} = -F'_j(\theta_k)/F''_{jj}(\theta_k)$ since b_k is excluded from both spatial and temporal regularization. Given δ_k's computed above, our algorithm iteratively updates θ_k's until there is no reduction on the overall objective function in Eq.(5.10). The final smoothly developed subnetworks are found by matching the optimal θ_k's with the graph topology of $\mathcal{G}^{(k)}$'s defined in Def. 2.3.

Algorithm Complexity: We name our algorithm `SLR` (Subnetwork Learning with Regularization) and analyze its complexity as follows. First, in terms of network topology, each network $S^{(i)}$ can be viewed as a subgraph of $\mathcal{G}^{(k)}$. Without loss of generality, we assume m as the maximum number of $\mathcal{G}^{(k)}$'s vertices. The computation of the log likelihood term therefore takes $O(Knm)$ time while that of the spatial smoothness takes $O(Km^2)$ (see Eq.(5.10)). The calculation of temporal smoothness takes $O(Km)$, whereas the 1st and 2nd derivatives of $F(\Theta, b)$ take $O(Knm)$ and $O(Km^2)$ respectively. These quantities are computed at each iteration of the gradient descent and thus the overall computation is $O(J(Km^2 + nm))$ with J as the number of iterations. Compared to m, both K and J are very small in practice. More importantly, the computation is not always quadratic in m since the subnetworks' size greatly reduces after each iteration due to the sparseness imposed on θ_k's.

6 Experiments

We compare `SLR` against the following techniques: (1) The well-known `MRMR` framework [20] that views network samples as collections of edges and performs edge selection based on mutual information; (2) `SSVM` by first applying SVD for dimensionality reduction (retaining 95% of the singular values) followed by SVM; (3) A recent typical graph classification method [4], named `GrphCls`, that works in a supervised setting and without side view information. We present two sets of experiments. First, to understand the strengths and limitations of our method, we utilize synthetic datasets that allow us to perform a number of controlled experiments. Second, we test all algorithms on the human functional brain networks to evaluate their practicability. Performance of every algorithm is evaluated via 5-fold stratified cross validation, in which their optimal parameters are chosen based on the estimated prediction accuracy within every 4 training folds and tested on the left-out fold according to [7]. Unless otherwise specified, λ_1 and λ_2 in `SLR` are selected from the ranges: $[0.001 - 50]$ with logscale step while with `GrphCls` [4], its *min-sup* is chosen from the range $[0.2 - 0.5]$ with step of 0.05. For `MRMR`, we use the forward edge selection scheme.

6.1 Data with known ground truth Following the approach described in [19,24], the synthetic datasets are generated by summary statistics associated with edges and adding ground truth signals at predefined

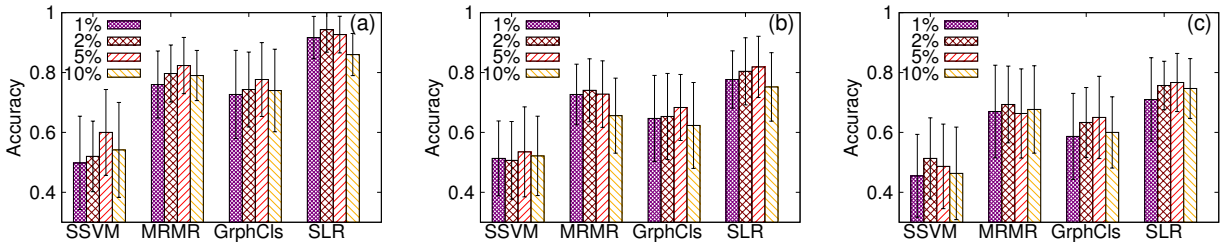

Fig. 2: Prediction accuracy of all techniques on synthetic datasets. Plots (a)-(c) correspond to CNR=$[1.5, 1.0, 0.5]$ respectively of all algorithms with GndSNet=$[1\%, 2\%, 5\%, 10\%]$ (std.dev. is shown on top of each bar).

subnetwork regions. The distribution of edges' values within the ground truth regions are generated with spatially contiguous correlations and conform to the background network structure. We use the contrast-to-noise (CNR) [19,24] to control the difference between ground truth and non-ground truth edges. Three batches of datasets are generated with CNR=$[1.5, 1.0, 0.5]$. Within each CNR setting, we further vary the percentage of edges (forming ground truth subnetworks) GndSNet=$[1\%, 2\%, 5\%, 10\%]$ of the total network edges, and the size of a ground truth subnetwork is varied between 3 to 15 edges each. In total, 12 datasets have been generated and their network samples are labeled to three global states. In simulation the evolving local network processes, ground truths for network samples labeled by global state 1 are firstly created, then the ground truths for networks with global state 2 are generated with the varying overlapping $[60\% - 90\%]$ with the ones in state 1. In a similar way, we generate the ground truths for network samples with global state 3 based on the ones in state 2. For each individual dataset, we generate 3K network samples evenly distributed into three global network states. Their network structures form approximately 6K vertices and 125K links within each graph $\mathcal{G}^{(k)}$ in Def.2.3.

Prediction Accuracy: Fig.2(a)-(c) shows the experimental results of `SLR` and the other three competing methods in predicting the global network states. Each plot in the figure corresponds to a CNR setting, and each bar corresponds to a setting of GndSNet. The accuracy values are obtained by averaging the prediction rate from cross validation. As one observes, `SLR` performs stably over variation in both CNR and GndSNet. Its prediction performance is better than `MRMR` since the network information is fully explored, while is superior to the network-based `GrphCls` since the temporal smoothness further narrows down the searching space to a small set of relevant and stable predictive subnetworks. Among all techniques, `SSVM` is less successful and the possible reason is that when irrelevant edges are prevalent in the data, aggregating all edges to form a lower dimensional subspace might not lead to a satisfactory performance.

Ground truth subnetwork discovery: We compute ROC curves based on the set of edges retrieved from the selected substructures w.r.t. the ground truth subnetworks. This is plotted in Fig.3(a)-(c) for all algorithms except `SSVM` since its new features combine information from all edges. The results are reported for GndSNet=5% as performance at other settings shows similar trends. For edge ranking in our `SLR` method, we rely on the absolute values of the parametric vectors, whereas for `GrphCls`, we rank edges based on their aggregated frequency in the significant subgraphs [4]. In `MRMR`, the ranking is based on the order in which edges are incrementally selected. The ROC curves shown in all three plots demonstrate that `SLR` uncovers more relevant subnetworks than `GrphCls` and `MRMR` for small positive rates. As this rate increases, the behaviors of three techniques become similar since none of them can fully uncover all ground truths. However, the higher ROC performance clearly makes `SLR` a better candidate in identifying local subnetworks influencing global network properties, which is practically important since validating the relevance of a subnetwork is often costly in real applications.

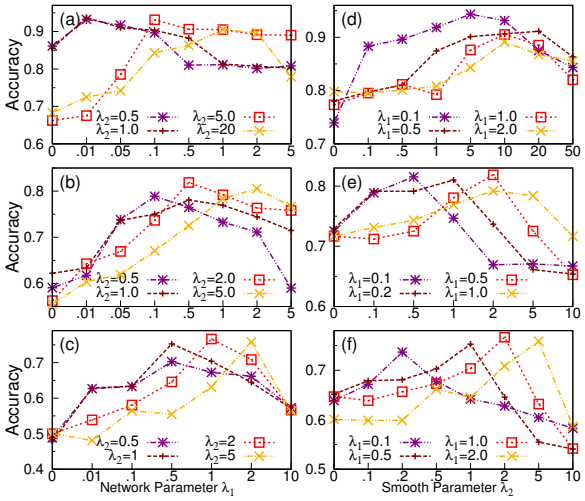

Fig. 4: Impact of λ_1 and λ_2 on prediction accuracy. Plots in three rows correspond to CNR=$[1.5, 1.0, 0.5]$ respectively with GndSNet=5%. Plots (a)-(c) show the impact of λ_1 while plots (d)-(f) show the impact of λ_2.

Impact of spatial and temporal network regularization: To provide more insights into the performance of our algorithm, we further report a series of experiments in examining the impact of λ_1 and λ_2. The intrinsic relationship between these two factors and `SLR`'s prediction

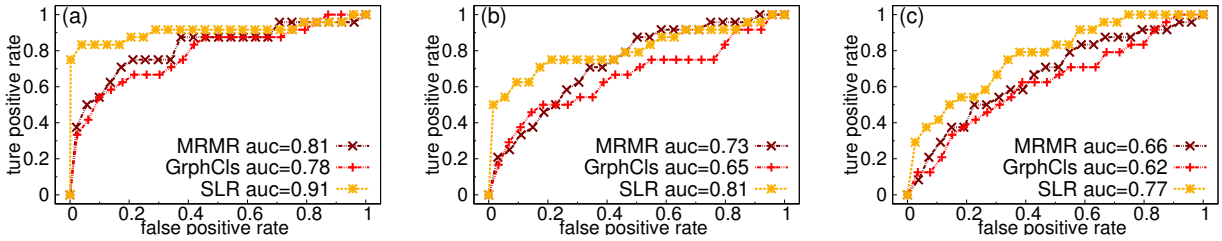

Fig. 3: ROC performance of the algorithms in uncovering the ground truth subnetworks. Plots (a)-(c) correspond to 3 levels of CNR=[1.5, 1.0, 0.5] with GndSNet=5% (similar trends were observed for other settings of GndSNet).

rate is plotted in Fig.4. In Fig.4(a-c), we show the impact of λ_1 on prediction rate when fixing λ_2 at different settings. An important trend can be seen that either setting λ_1 too small or large do not lead to a high prediction rate. A small λ_1 causes isolated edges to be selected since the network topology is disregarded, while a large λ_1 can overly favor strongly connected substructures yet not related to the global network states. A similar trend is also seen in Figure 4(d-f) where we fix λ_1 and vary λ_2. A small setting of λ_2 causes different subnetworks across different global network states, while its large value may force them to be too similar across states, even for non-ground truth substructures, both leading to a low prediction accuracy.

6.2 Real world dataset We choose the important application of analyzing human brain networks associated with the Alzheimer's Disease. The analysis of brain data has recently attracted much attention from the data mining community with convincing results demonstrated in [4,5,17,28]. However, unlike most previous studies which focus on a small number of subjects and especially not for the *temporal* development of the disease, we analyze a large scale cohort of 180 subjects obtained from http://www.adni-info.org/, and evenly distributed into three global states: normal control (NC), mild cognitive impairment (MCI) and Alzheimer's disease (AD). FSL toolbox [23] is used to convert an fMRI scan to a functional brain network, comprising of 112 brain regions. A value associated with an edge (connecting two brain regions) is evaluated by the correlation between their blood oxygen level-dependent time series. Since there is no gold standard for choosing a proper threshold for functional correlation, we follow the general approach in [22,25] by selecting four thresholds ranging from $[0.8, 0.4, 0.2, 0]$ to remove weak correlations, resulting in four network datasets respectively denoted by `C8`, `C4`, `C2` and `C0`, with numbers of vertices/links varying from 776/15535 to 5515/49284.

Prediction Performance: We evaluate all algorithms on network state prediction by comparing their performance on five settings of selected subnetworks between 1%, 2%, 5% 10% and 12% of total unique edges. Fig.5 reveals that all algorithms yield better prediction rates

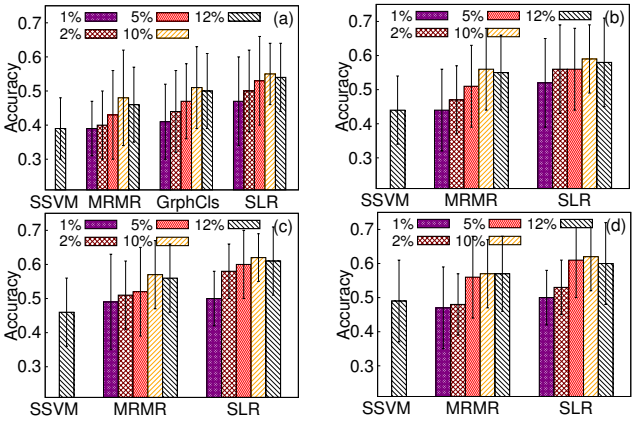

Fig. 5: Prediction accuracy on human brain networks. Plots (a)-(d) correspond to datasets `C8`, `C4`, `C2` and `C0` respectively, with the selected subnetworks varied from 1%, 2%, 5%, 10% to 12% of the total edges. `GrphCls` does not handle well densely connected datasets, while `SSVM` is shown with a single column due to using aggregated features.

for higher numbers of selected substructures. However, a level larger than 10% does not lead to better prediction accuracy, due to the prevalence of noisy edges. Among all examined techniques, `SSVM` shows the lowest performance at the prediction rate of 39% in `C8`, a result that is only marginally better than the random guess of 33%. `GrphCls` performs much better than `SSVM` but only works on the sparse network data `C8`. For other datasets with denser network samples, `GrphCls` takes much longer time handle the large amount of possible substructures. The performance of `SLR` is by far the best, dominating both mutual information based `MRMR` and the frequent subgraph-based `GrphCls`, with a large accuracy advantage on both sparsely and densely connected network data. Looking deeper, we also see that for the same level of prediction rate, subnetworks uncovered by `SLR` is usually more succinct compared to other competing methods. For example, on the densely connected dataset `C2`, it achieves 60% of prediction accuracy with 5% of total edges combined from its substructures, as compared to the second-best technique `MRMR`, which requires double the number of edges for a prediction rate of 57%.

Subnetwork discovery: We evaluate the quality of uncovered substructures from each technique through their consistency in cross validation. Subnetworks that are

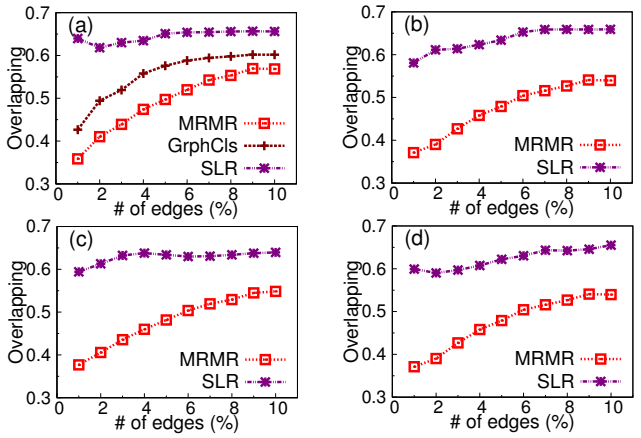

Fig. 6: Overlapping percentage of selected edges across all training data. Plots (a)-(d) correspond to four network datasets C8, C4, C2 and C0.

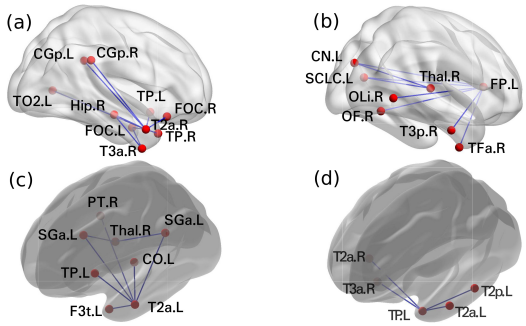

Fig. 7: Four typical subnetwork markers (visualized with node hubs) consistently selected by SLR. Nodes are brain regions abbreviated in the HO atlas standard [23]. (a) T2a.R: Right Middle Temporal Gyrus, anterior division; (b) T2a.L: Left Middle Temporal Gyrus, anterior division; (c) FP.L: Left Frontal Pole; (d) TP.L: Left Temporal Pole.

consistently selected across training folds are likely the disease-related biomarkers and they should be the first candidate for further investigation. Fig.6 provides overlapping percentage (y-axis) of discovered subnetworks across all training folds for the four datasets. As observed, although the overlap tends to increase as the number of selected edges increases, none of the competing methods consistently produce results as good as SLR, which implies that SLR selects the most stable and consistent predictive substructures across all training folds.

To validate that our proposed method finds meaningful subnetworks, we further investigate its selected subnetworks. Fig.7 displays the top four subnetworks consistently discovered from all training folds. For easier visualization, we plot each subnetwork with a core node that has the highest node degree (like community hub [21]). It is found that the core nodes like T2a.R, T2a.L and TP.L indeed reside within the temporal lope—the brain region strongly impacted by Alzheimer's disease as reported in [18,26].

Fig.8 further provides visualization on how these subnetworks have changed smoothly across the global network states. Three columns in each plot correspond to NC, MCI and AD states. As seen, functional correlations deteriorate noticeably from NC to AD groups in most cases, especially those in the T2a.R subnetwork (Fig. 8(a)) and the TP.L subnetwork (Fig. 8(d)). This phenomenon can be explained by the fact that, once T2a.R and TP.L regions are damaged by the disease, their cognitive performance is significantly reduced which further impacts other functionally connected regions. However, we also observe that in some circumstances, the functional correlation increases from NC to AD, as between FP.L and SCLC.L, OLi.R in Fig. 8(b), or between Thal.R and CGa.L, SGa.L in Fig. 8(c). This increased functional connectivity might support the "compensatory recruitment hypothesis" [18].

Scalability: We also evaluate the scalability of our

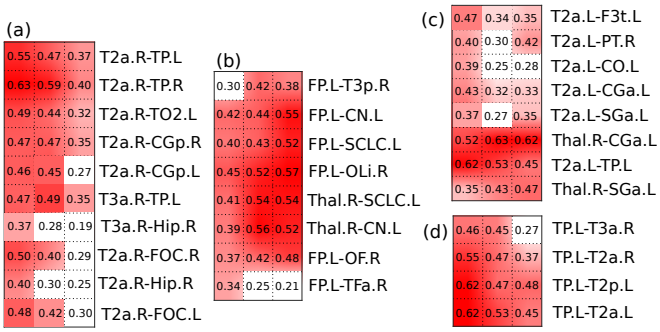

Fig. 8: Smoothly changing subnetworks. Columns from left to right in each plot correspond to NC, MCI and AD states. Dark background colors show selected edges whereas entries' values show functional correlations averaged from network samples within the corresponding group.

algorithm against other methods. To provide an idea about typical running time: SLR takes 241s, 248s, 279s and 292s on four datasets from C8 to C0. These numbers with the MRMR are respectively 20s, 198s, 4686s and 5055s while with the GraphCls is 3669s on C8 dataset (for a setting of subnetworks with 5% total edges). SLR shows better scalability. Its computation overhead is quadratic in initial iterations but significantly reduces in subsequent iterations since irrelevant substructures get zero values in the parametric vectors.

7 Related work

Analyzing network structural data has been widely studied in the literature with most existing work focusing on community detection [21], frequent subgraph mining [10], outlier detection [2], and graph classification [11,12,27]. Close to our study is the line of work on graph classification. Though diverse in terms of underlying approaches, most algorithms [11,27] generally assume a database consisting of "positive" and "negative" graphs, and aim at extracting a set significant subgraphs that are frequently *present* in one class but *absent* in the other, which subsequently are used as new

(binary) features to train classifiers. These approaches, recently being extended to semi-supervised setting [16], uncertain graphs [15], or multiple side-view [4] which we have adapted for our empirical comparison. Another related work is the one developed in [28] that directly addresses the progressive data but in the *non-network* context. Moreover, its view on the subject behaviors' progression differs from ours in which it assumes the smooth changes in predictive variables appear in every model while minimizing the models' difference learnt at various time points. In contrast, our study explores a single model in which the changes in local network processes can lead to the changes in global network states.

Another line of related studies are from feature selection where one can view each network sample as a collection of edges and use statistical analysis t-test [13,17] or mutual information [6,20] to select edges that lead to the statistical difference among various global network states. The advantage is that these studies can make an individual edge or region-based analysis [6,13], instead of analyzing entire networks as a whole. However, they lack the capability of analyzing both intra and inter-connectivities among different network regions at the same time. Our work is also related to dynamic network mining that aims to discover subnetworks of interest in multiple discrete snapshots of an evolving network [14] or in a multi-layer network [8]. The goal in this line of work, however, is not predicting global network states, but the discovery of abnormal substructures that persist in time or across network layers.

8 Conclusions

In this paper, we address an important problem of mining a succinct set local subnetworks that are predictive for the progression of global network states. We develop `SLR` as a novel algorithm that fits a model for multi-states of network samples subject to two important constraints: (i) spatial smoothness imposed on the network topology to ensure the well-connected substructures; (ii) temporal smoothness to discover predictive subnetworks evolving along with the progression of global network states. `SLR` further imposes the sparsity-inducing $L1$-norm to explicitly remove edges that have little or no impact on the progression of global network states and we show that the overall optimization function is convex. Extensive experiments on both synthetic datasets and the emerging brain networks demonstrate the appealing performance of our algorithm not only in terms of prediction accuracy but also in the consistency of the discovered subnetworks with existing literature, providing a better understanding of the intrinsic relationship between the evolution of local network processes and the progression of global network behaviours.

References

[1] Alzheimer's disease facts and figures. *Alzheimer's & Dementia*, 11(3), 2015.

[2] L. Akoglu et al. Graph based anomaly detection and description: a survey. *DMKD*, 2015.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[4] B. Cao et al. Mining brain networks using multiple side-views for neurological disorder identification. In *ICDM*, 2015.

[5] I. N. Davidson et al. Network discovery via constrained tensor analysis of fMRI data. In *SIGKDD*, 2013.

[6] E. L. Dennis et al. Functional brain connectivity using fMRI in aging and Alzheimer's. *Neuropsychol Rev*, 2014.

[7] T. Hastie et al. *The Elements of Statistical Learning. Data Mining, Inference, and Prediction*. 2009.

[8] P. Holme and J. Saramäki. Temporal networks. *Physics Reports*, 519(3):97–125, 2012.

[9] G. Isabelle and E. André. An introduction to variable and feature selection. *JMLR*, 2003.

[10] C. Jiang et al. A survey of frequent subgraph mining algorithms. *Knowledge Eng. Review*, 28(1), 2013.

[11] N. Jin et al. GAIA: graph classification using evolutionary computation. In *SIGMOD*, 2010.

[12] N. S. Ketkar et al. Empirical comparison of graph classification algorithms. In *CIDM*, 2009.

[13] J. Kim et al. Comparison of statistical tests for group differences in brain functional networks. *NeuroImage*, 2014.

[14] M. Kivelä, Arenas, et al. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 2014.

[15] X. Kong et al. Discriminative feature selection for uncertain graph classification. In *SIAM-SDM*, 2013.

[16] X. Kong and P. Yu. Semi-supervised feature selection for graph classification. In *SIGKDD*, 2010.

[17] X. Kong and P. Yu. Brain network analysis: a data mining perspective. *SIGKDD Explorations*, 2013.

[18] E. W. Lang et al. Brain connectivity analysis: A short survey. *Intell. Neuroscience*, 2012, Jan. 2012.

[19] F. Martino et al. Combining multivariate voxel selection and support vector machines for mapping and classification of fMRI spatial patterns. *NeuroImage*, 2008.

[20] H. Peng et al. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE TPAMI*, 27(8), 2005.

[21] S.Harenberg. Community detection in large-scale networks: a survey and empirical evaluation. *Comp. Stat.*, 2014.

[22] S. L. Simpson et al. Analyzing complex functional brain networks: Fusing statistics and network science to understand the brain. *Statistical Surveys*, 7, 2013.

[23] S. Smith. Advances in functional and structural MRI analysis and implementation as FSL. *Neuroimage*, 2004.

[24] S. M. Smith et al. Network modelling methods for fMRI. *NeuroImage*, 54(2):875 – 891, 2011.

[25] O. Sporns. Networks analysis, complexity, and brain function. *Complex.*, 8(1):56–60, Sept. 2002.

[26] K. Supekar et al. Network analysis of intrinsic functional brain connectivity in alzheimer's disease. *PLoS*, 2008.

[27] X. Yan et al. Mining significant graph patterns by leap search. In *SIGMOD*, 2008.

[28] J. Zhou et al. A multi-task learning formulation for predicting disease progression. In *SIGKDD*, 2011.