Published in final edited form as:
KDD. 2011 ; : 931–939. doi:10.1145/2020408.2020562.

Brain Effective Connectivity Modeling for Alzheimer's Disease by Sparse Gaussian Bayesian Network

Shuai Huang[1], **Jing Li**[1], **Jieping Ye**[1], **Adam Fleisher**[2], **Kewei Chen**[2], **Teresa Wu**[1], and **Eric Reiman**[2]

[1]School of Computing, Informatics, and Decisions Systems Engineering, Arizona State University, Tempe, AZ, 85287

[2]Banner Alzheimer's Institute, Banner Good Samaritan Medical Center, Phoenix, AZ, 85006

Abstract

Recent studies have shown that Alzheimer's disease (AD) is related to alteration in brain connectivity networks. One type of connectivity, called effective connectivity, defined as the directional relationship between brain regions, is essential to brain function. However, there have been few studies on modeling the effective connectivity of AD and characterizing its difference from normal controls (NC). In this paper, we investigate the sparse Bayesian Network (BN) for effective connectivity modeling. Specifically, we propose a novel formulation for the structure learning of BNs, which involves one L1-norm penalty term to impose sparsity and another penalty to ensure the learned BN to be a directed acyclic graph – a required property of BNs. We show, through both theoretical analysis and extensive experiments on eleven moderate and large benchmark networks with various sample sizes, that the proposed method has much improved learning accuracy and scalability compared with ten competing algorithms. We apply the proposed method to FDG-PET images of 42 AD and 67 NC subjects, and identify the effective connectivity models for AD and NC, respectively. Our study reveals that the effective connectivity of AD is different from that of NC in many ways, including the global-scale effective connectivity, intra-lobe, interlobe, and inter-hemispheric effective connectivity distributions, as well as the effective connectivity associated with specific brain regions. These findings are consistent with known pathology and clinical progression of AD, and will contribute to AD knowledge discovery.

General Terms

Algorithms

Keywords

Brain network; Alzheimer's disease; neuroimaging; FDG-PET; Bayesian network; sparse learning

1. INTRODUCTION

Alzheimer's disease (AD) is the most common cause of dementia and the fifth leading cause of death in people over 65 in the US. The current annual cost of AD care in the U.S. is more than $100 billion, which will continue to grow fast. The existing knowledge about the cause of AD is very limited. Clinical diagnosis is imprecise with a definite diagnosis only possible by autopsy. Also, there is currently no cure for AD, while most drugs only modestly alleviate symptoms. To tackle these challenging issues in AD studies, fast advancing neuroimaging techniques hold great promise. Recent studies have shown that neuroimaging can provide sensitive and reliable measures of AD onset and progression, which can complement the conventional clinical-based assessments and cognitive measures.

In neuroimaging-based AD research, one important area is brain connectivity modeling, i.e., identification of how different brain regions interact to produce a cognitive function in AD, compared with normal aging. Research in this area can substantially promote AD knowledge discovery and identification of novel connectivity-based AD biomarkers to be used in clinical practice. There are two types of connectivity being studied: *functional connectivity* refers to the covarying pattern of different brain regions; *effective connectivity* refers to the directional relationship between regions [1].

A vast majority of the existing research focuses on functional connectivity modeling. Various methods have been adopted such as correlation analysis [2], Principal Component Analysis (PCA) [3], PCA-based Scaled Subprofile Model [4], Independent Component Analysis [5], and Partial Least Squares [6]. Recently, sparse models have also been introduced, such as sparse multivariate or vector autoregressions [7] and sparse inverse covariance estimation [8]. Sparse models have shown great effectiveness because neuroimaging datasets are featured by "small n large p", i.e., the number of AD patients (n) can be close to or less than the number of brain regions modeled (p). Also, many past studies based on anatomical brain databases have shown that the true brain network is indeed sparse [9].

Compared with functional connectivity modeling, effective connectivity modeling has the advantage of helping identify the pathway/mechanism whereby distinct brain regions communicate with each other. However, the existing research in effective connectivity modeling is much less extensive. Models that have been adopted include structural equation models [10] and dynamic causal models [11]. The limitations of these models include (i) they are confirmative, rather than explanatory, i.e., they require a prior model of connectivity to start with; (ii) they require a substantially larger sample size than the number of regions modeled; a typical number of regions included is less than 10 given the sample size limit. These limitations make them inappropriate for AD connectivity modeling, because there is little prior knowledge of which regions should be included and how they are connected.

We propose sparse Bayesian Networks (BN) for effective connectivity modeling. A BN is an explanatory model and the sparse estimation makes it possible to include a large number of brain regions. In a BN representation of effective connectivity, the nodes are brain regions. A directed arc from node X_i to X_j (X_i is called a parent of X_j) indicates a direct influence from X_i to X_j. Sparsity consideration has been common in the BN learning literature. For example, some early work used score functions, such as BIC and MDL, to measure the goodness-of-fit of a BN, in which a penalty term on the model complexity is usually included in the score [12]. Recently, driven by modern applications such as genetics, learning of large-scale BNs has been very popular, in which sparsity consideration is indispensible. The Sparse Candidate (SC) algorithm [13], one of the first BN structure learning algorithms to be applied to a large number of variables, assumes that the maximum number of parents for each node is limited to a small constant. The L1MB-DAG algorithm developed in [14] uses LASSO [15] to identify a small set of potential parents for each variable. Some other algorithms, such as the Max-Min Hill-Climbing (MMHC) [16], Grow-Shrink [17], TC and TC_bw [18], all follow the same line. Most of these existing algorithms employ a two-stage approach: Stage 1 is to identify the potential parents of each variable; Stage 2 usually applies some heuristic search algorithms (e.g., hill-climbing) or orientation methods (e.g., Meek's rules [19]) to identify the parents out of the potential parent set. An apparent weakness of the two-stage approach is that if a true parent is missed in Stage 1, it will never be recovered in Stage 2. Another weakness of the existing algorithms is computational efficiency, i.e., it may take hours or days to learn a large-scale BN such as one with 500 nodes.

In this paper, we propose a new sparse BN learning algorithm, which is called SBN. It is a one-stage approach that identifies the parents of all variables directly. The main contributions of this paper include:

- We propose a novel sparse BN learning algorithm, i.e., SBN, using one L1-norm penalty term to impose sparsity and another penalty term to ensure the learned BN is a directed acyclic graph (DAG).
- We present theoretical guidance on how to select the regularization parameter associated with the second penalty.
- We perform theoretical analysis to reason why the two-stage approach popularly adopted in the existing literature has a high risk of failing to identify the true parents. Also, we conduct extensive experiments on synthetic data to compare SBN and the existing algorithms in terms of the learning accuracy and scalability.
- We apply SBN to FDG-PET data of 42 AD patients and 67 normal controls (NC) subjects enrolled in the ADNI (Alzheimer's Disease Neuroimaging Initiative) project, and identify the effective connectivity models for AD and NC. Our study reveals that the effective connectivity of AD is different from NC in many ways, including the global-scale effective connectivity, intra-lobe, inter-lobe, and inter-hemisphere effective connectivity distributions, as well as the effective connectivity associated with specific brain regions. The findings are consistent with known pathology and clinical progression of AD.

2. BAYESIAN NETWORK: KEY DEFINITIONS AND CONCEPTS

This section introduces the key definitions and concepts of BNs that are relevant to this paper:

A BN is composed of a structure and a set of parameters. The structure (Fig. 1) is a DAG that consists of p nodes $[X_1, \ldots, X_p]$ and directed arcs between some nodes; no cycle is allowed in a DAG. Each node represents a random variable. If there is a directed arc from X_i to X_j, X_i is called a *parent* of X_j and X_j is called a *child* of X_i Two nodes are called *spouses* if they share a common child. If there is a *directed path* from X_i to X_j, i.e., $X_i \rightarrow \cdots \rightarrow X_j$, X_i is called an *ancestor* of X_j. A directed arc is also a directed path and a parent is also an ancestor according to this definition. The *Markov Blanket (MB)* of X_j is a set of variables given which X_j will be independent of all other variables. The MB includes the parents, children, and spouses of X_j.

In this paper, we use the following notations with respect to a BN structure: Denote the structure by a $p \times p$ matrix $\mathbf{G}$, with entry $\mathbf{G}_{ij} = 1$ representing a directed arc from X_i to X_j. The set of parents of a node X_i is denoted by $\mathbf{PA}(X_i) = [X_{i_1}, \ldots, X_{i_k}]^T$. In addition, we define a $p \times p$ matrix, $\mathbf{P}$, which records all the directed paths in the structure, i.e., if there is a directed path from X_i to X_j, entry $\mathbf{P}_{ij} = 1$; otherwise, $\mathbf{P}_{ij} = 0$.

In addition to the structure, other important components of a BN are the parameters. The parameters are the conditional probability distribution of each node given its parents. Specifically, when the nodes follow a multivariate Gaussian distribution, a regression-type parameterization can be adopted, i.e., $X_i = \beta_i^T \mathbf{PA}(X_i) + \varepsilon_i$ with $\varepsilon_i \sim N(0,\sigma_i^2)$ and $\beta_i = [\beta_{i_1 i},\cdots, \beta_{i_k i}]^T$. Then, the parameters of a BN are $\mathbf{B} = [\beta_1, \ldots, \beta_p]$. Without loss of generality, we assume that the nodes are standardized, i.e., each with a zero mean and unit variance. This means, if using $\boldsymbol{x}_i = [x_{i1}, \ldots, x_{in}]$ to denote the sample vector for X_i, and n to denote the sample size, we have $\sum_{j=1}^{n} x_{ij}=0$ and $\boldsymbol{x}_i\boldsymbol{x}_i^T=n-1$.

3. THE PROPOSED SPARSE BN STRUCTURE LEARNING ALGORITHM

One of the challenging issues in BN structure learning is to ensure that the learned structure must be a DAG, i.e., no cycle is present. To achieve this, we first identify a sufficient and necessary condition for a DAG, which is given as Lemma 1 below.

Lemma 1. *A sufficient and necessary condition for a DAG is* $\beta_{ji} \times \mathbf{P}_{ij} = 0$ *for every pair of nodes* X_i and X_j.

Proof. To prove the necessary condition, suppose that a BN structure, $\mathbf{G}$, is a DAG. Let's assume that $\beta_{ji} \times \mathbf{P}_{ij} \neq 0$ for a pair of nodes X_i and X_j. Then, there exists a directed path from X_j to X_i and a directed path from X_i to X_j, i.e., there is a cycle in $\mathbf{G}$, which is a contradiction to our presumption that $\mathbf{G}$ is a DAG. To prove the sufficient condition, suppose that $\beta_{ji} \times \mathbf{P}_{ij} = 0$ for every pair of nodes X_i and X_j. If $\mathbf{G}$ is not a DAG, i.e., there is a cycle, it means that there exist two variables, X_i and X_j, with a directed arc from X_j to X_i ($\beta_{ji} \neq 0$) and a directed path from X_i to X_j ($\mathbf{P}_{ij} = 1$). This contradicts with our assumption that $\beta_{ji} \times \mathbf{P}_{ij} = 0$ for every pair of nodes X_i and X_j.

Based on Lemma 1, we further present our formulation of the sparse BN structure learning. It is an optimization problem with the objective function and constraints given by:

$$\hat{\mathbf{B}}{=}min_{\mathbf{B}}\sum_{i=1}^{p}\left\{(\boldsymbol{x}_i-\boldsymbol{\beta}_i^T\boldsymbol{x}_{/i})(\boldsymbol{x}_i-\boldsymbol{\beta}_i^T\boldsymbol{x}_{/i})^T{+}\lambda_1\left\|\boldsymbol{\beta}_i\right\|_1\right\} s.t.\beta_{ji}{\times}\mathbf{P}_{ij}{=}0, i, j{=}1, \ldots, p, i \neq j. \quad (1)$$

The notations are explained as follows: $\boldsymbol{x}_i = [x_{i1}, \ldots, x_{in}]$ denotes the sample vector for X_i, where n is the sample size. $\boldsymbol{x}_{/i}$ denotes the sample matrix for all the variables except X_i .The first term in the objective function, $\sum_{i=1}^{p}\left\{(\boldsymbol{x}_i-\boldsymbol{\beta}_i{}^T\boldsymbol{x}_{/i})(\boldsymbol{x}_i-\boldsymbol{\beta}_i{}^T\boldsymbol{x}_{/i})^T\right\}$, is a profile likelihood to measure the model fit. The second term,$\|\beta_i\|_1$, is the sum of the absolute values of the elements in β_i and thus is the so-called L1-norm penalty [15]. The regularization parameter, λ_1, controls the number of nonzero elements in the solution to β_i, $\hat{\beta}_i$; larger λ_1, less nonzero elements. Because less nonzero elements in $\hat{\beta}_i$ correspond to fewer arcs in the learned BN structure, a larger λ_1 results in a sparser structure. In addition, the constraints are to assure that the learned BN is a DAG (see Lemma 1 and Theorem 1 below). We remark that these constraints are functions of **B** only, since **P** = expm(**G**) [20]. Here, expm(**G**) is the matrix exponential of **G**.

Solving the constrained optimization in (1) is difficult. Therefore, the penalty method [21] is employed to transform it into an unconstrained optimization problem, through adding an extra L1-norm penalty into the original objective function, i.e.,

$$\hat{\mathbf{B}}_{ap}{=}\min_{\mathbf{B}}\sum_{i=1}^{p}f_i(\boldsymbol{\beta}_i){=}min_{\mathbf{B}}\sum_{i=1}^{p}\left\{(\boldsymbol{x}_i-\boldsymbol{\beta}_i^T\boldsymbol{x}_{/i})(\boldsymbol{x}_i-\boldsymbol{\beta}_i^T\boldsymbol{x}_{/i})^T{+}\lambda_1\|\boldsymbol{\beta}_i\|_1{+}\lambda_2\sum_{j\in X_{/i}}|\beta_{ji}\times\mathbf{P}_{ij}|\right\}, \quad (2)$$

where $j \in X_{/\mathrm{i}}$ denotes that the variable indexed by j, i.e., X_j is a variable different from X_i. Here, $\lambda_2 \sum_{j\in X_{/i}}|\hat{\beta}_{ji} \times \mathbf{P}_{ij}|$ is to push $\beta_{ji} \times \mathbf{P}_{ij}$ become zero. Under some mild conditions, there exists a λ_2^* such that for all $\lambda_2 \geq \lambda_2^*$, $\hat{\mathbf{B}}_{ap}$ is also a minimizer for (1) [21]. Theorem 1 gives a practical estimation for λ_2^*.

Theorem 1. *Any* $\lambda_2 > (\mathrm{n} - 1)^2\, p/\lambda_1 - \lambda_1$ *will guarantee* $\hat{\mathbf{B}}_{ap}$ *to be a DAG.*

Proof. To prove this we first need to show that with a certain value of λ_1 and any value of λ_2, $\hat{\mathbf{B}}_{\mathrm{ap}}$ is bounded. This can be seen from

$$\lambda_1\left\|\hat{\beta}_i\right\|_1 \leq \left(\boldsymbol{x}_i-\hat{\beta}_i{}^T\boldsymbol{x}_{/i}\right)\left(\boldsymbol{x}_i-\hat{\beta}_i{}^T\boldsymbol{x}_{/i}\right)^T{+}\lambda_1\left\|\hat{\beta}_i\right\|_1{+}\lambda_2\sum_{j\in X_{/i}}|\hat{\beta}_{ji}{\times}\mathbf{P}_{ij}| \leq \boldsymbol{x}_i\boldsymbol{x}_i^T{=}n{-}1,$$

for each $\hat{\beta}_i$. The last inequality holds because $\boldsymbol{x}_i\boldsymbol{x}_i^T$ is the value of the function on the left hand side with $\beta_i = 0$, which is obviously larger than the function value with $\beta_i.= \hat{\beta}_i$ The last equality holds since we have standardized all the variables. Thus, we know that $\max_{k\in\mathbf{PA}(X_i)}|\hat{\beta}_{ki}| \leq (n - 1)/\lambda_1$. Now, we use proof-by-contradiction to show that, with any $\lambda_2 > (n - 1)^2 p/\lambda_1 - \lambda_1$, we will get a DAG. Suppose that such a λ_2 doesn't guarantee a DAG. Then, there must be at least a pair of variables X_i and X_j with $\beta_{ji} \times \mathbf{P}_{ij} \neq 0$, i.e., $\beta_{ji} \neq 0$ and $\mathbf{P}_{ij}$

= 1. Based on the first order optimality condition, $\beta_{ji} \neq 0$ i.f.f. $\left|(\boldsymbol{x}_i - \hat{\beta}^T_{i/j}\boldsymbol{x}_{/(i,j)})\boldsymbol{x}^T_j\right| - (\lambda_1 + \lambda_2|\mathbf{P}_{ij}|) > 0$. Here, $\hat{\beta}^T_{i/j}$ denotes the elements in $\hat{\beta_i}$ without $\hat{\beta_{ji}}$, and $\boldsymbol{x}_{/(i,j)}$ denotes the sample matrix for all the variables except X_i and X_j. However, we have

$$|(\boldsymbol{x}_i - \hat{\beta}^T_{i/j}\boldsymbol{x}_{/(i,j)})\boldsymbol{x}^T_j| \leq |\boldsymbol{x}_i\boldsymbol{x}^T_j| + \sum\nolimits_{k \in X_{/(i,j)}} |\hat{\beta}_{ki}\boldsymbol{x}_k\boldsymbol{x}^T_j| < (n-1)p\max\nolimits_{k \in X_{/i}} \hat{\beta}_{ki} < (n-1)^2 p/\lambda_1,$$

which results in $\left|\left(\boldsymbol{x}_i - \hat{\beta}^T_{i/j}\boldsymbol{x}_{/(i,j)}\right)\boldsymbol{x}^T_j\right| - (\lambda_1 + \lambda_2\left|\mathbf{P}_{ij}\right|) < 0$

Theorem 1 implies that if we specify any $\lambda_2 > (n-1)^2 p/\lambda_1 - \lambda_1$, we will get a minimizer of (1) through solving (2). However, in practice, directly solving (2) by specifying a large λ_2 may converge slowly. This is because that the unconstrained problem in (2) may be ill-conditioned with a too large value for λ_2 [40]. To avoid this, the "warm start" method [21] can be used, which works in the following way: first, it specifies a series of values for λ_2, i.e., $\lambda^0_2 < \lambda^1_2 < \lambda^2_2 < \cdots < \lambda^M_2$, where λ^0_2 is small and $\lambda^M_2 > (n-1)^2 p/\lambda_1 - \lambda_1$; next, it optimizes (2) with $\lambda_2 = \lambda^0_2$ to get a minimizer $\hat{\mathbf{B}}^0_{ap}$, using an arbitrary initial value; then, it optimizes (2) with $\lambda_2 = \lambda^1_2$, using $\hat{\mathbf{B}}^0_{ap}$ as an initial value; this process iterates, until it optimizes (2) with $\lambda_2 = \lambda^M_2$. With the last minimizer as the initial value for the next optimization problem, this method can be quite efficient.

Given λ_1 and λ_2, the BCD algorithm [22] can be employed to solve (2). The BCD algorithm updates each $\boldsymbol{\beta_i}$ iteratively, assuming that all other parameters are fixed. In our situation, this is equivalent to optimizing $f_i(\boldsymbol{\beta_i})$ iteratively and the algorithm will terminate when some convergence conditions are satisfied. We remark that $f_i(\boldsymbol{\beta_i})$, after some transformation, is similar to LASSO [15], i.e.,

$$f_i(\boldsymbol{\beta}_i) = (\boldsymbol{x}_i - \boldsymbol{\beta}^T_i\boldsymbol{x}_{/i})(\boldsymbol{x}_i - \boldsymbol{\beta}^T_i\boldsymbol{x}_{/i})^T + \sum\nolimits_{j \in X_{/i}} (\lambda_1 + \lambda_2|\mathbf{P}_{ij}|)|\beta_{ji}|. \quad (3)$$

As a result, the shooting algorithm [23] for LASSO may be used to optimize $f_i(\boldsymbol{\beta_i})$ in each iteration. Note that at each iteration for optimizing $f_i(\boldsymbol{\beta_i})$, we also need to calculate $\mathbf{P}_{ij}$ for $j \in X_{/i}$. This can be done by a Breadth-first search on $\mathbf{G}$ with X_i being the root node. A more detailed description of the BCD algorithm used to solve (2) is given in Figure 2.

Finally, we want to mention that the L2-norm penalty, $\lambda_2 \sum_{j \in X_{/i}} (\beta_{ji} \times \mathbf{P}_{ij})^2$, might also be used in (2). The advantage is that it is a differentiable function of β_{ji}. Also, as shown in [21], $\beta_{ji} \times \mathbf{P}_{ij} \to 0$ when $\lambda_2 \to \infty$. However, the limitation of the L2-norm penalty, compared with the L1-norm penalty, is that there is no guarantee that a finite λ_2 exists to assure $\beta_{ji} \times \mathbf{P}_{ij} = 0$ for all pairs of X_i and X_j.

Time complexity analysis of the proposed algorithm: Each iteration of the BCD algorithm consists of two operations: a shooting algorithm and a Breadth-first search on **G**. These two operations cost $O(pn)$ [24] and $O(p + |\mathbf{G}|)$, respectively. Here $|\mathbf{G}|$ is the number of nonzero elements in **G**. If **G** is sparse, i.e., $|\mathbf{G}| = C_p$ with a small constant C, $O(p + |\mathbf{G}|) = O(p)$. Thus, the computational cost at each iteration is only $O(pn)$. Furthermore, each sweep through all columns of **B** costs $O(p^2n)$. Our simulation study shows that it usually takes no more than 5 sweeps to converge.

4. THEORETICAL ANALYSIS OF THE COMPETITIVE ADVANTAGE OF THE PROPOSED SBN ALGORITHM

Simulation studies in Sec. 5 will show that SBN is more accurate than various existing algorithms that employ a two-stage approach. This section aims to provide some theoretical insights about why it is so. Recall that Stage 1 of the two-stage approach is to identify potential parents of each variable X_i. The existing algorithms achieve this by identifying the MB of X_i. A typical way is variable selection based on regression, i.e., to build a regression of X_i on all other variables and consider the variables selected to be the MB. The key differences between various algorithms are the type of regression used and the method in variable selection. For example, the TC algorithm [18] uses ordinary regression and a t-test for variable selection; the L1MB-DAG algorithm [14] uses LASSO.

In the regression of X_i, the coefficients for the variables that are not in the MB will be small (theoretically zero due to the definition of MB). However, the coefficients for the parents may also be very small due to the correlation between the parents and children. As a result, some parents may not be selected in the variable selection, i.e., they will be missed in Stage 1 of the two-stage approach, leading to greater BN learning errors. In contrast, the SBN may not suffer from this, because it is a one-stage approach that identifies the parents directly.

To further illustrate this point, we analyze one two-stage algorithm, the TC algorithm. TC does variable selection using a t-test. To determine whether a variable should be selected, a t-test uses the statistic $\hat{\beta}/se(\hat{\beta})$, where $\hat{\beta}$ is the estimate for the regression coefficient of this variable and $se(\hat{\beta})$ is the standard error. The larger the $\hat{\beta}/se(\hat{\beta})$, the higher chance the variable will be selected. Theorems 2 and 3 below show that even though the value of $\hat{\beta}/se(\hat{\beta})$ corresponding to a parent of X_i is large in the true BN, its value may decrease drastically in the regression of X_i on all other variables. Theorem 2 focuses on a specific type of BNs, a general tree, in which all variables have one common ancestor and there is at most one directed path between two variables. Theorem 3 focuses on a general inverse tree, which becomes a general tree if all the arcs are reversed. The proof of Theorem 2 can be found in the Appendix. The proof of Theorem 3 is not shown here due to space limitations.

Theorem 2. Consider a general tree with $m + 2$ variables, whose structure and parameters are given by $X_1 = e_1$, $X_2 = \beta_{12}X_1 + e_2$, $X_i = \beta_{2i}X_2 + e_i$, $i = 3,4, \ldots, m$. All the variables have unit variance. If X_2 is regressed on all other variables in Stage 1 of TC, i.e., $X_2=\beta_{12}^{MB}X_1+\beta_{23}^{MB}X_3+\cdots+\beta_{2m}^{MB}X_n+e_2^{MB}$, the coefficient for X_2's parent, X_1, is:

$$\beta_{12}^{MB}{=}\beta_{12}\frac{\prod_{i=3}^{m}(1-\beta_{2i}^{2})}{\prod_{i=3}^{m}(1-\beta_{2i}^{2}){+}\sum_{i=3}^{m}[\beta_{2i}^{2}\prod_{j=3,j\neq i}^{m}(1-\beta_{2i}^{2})]}{<}\beta_{12},\text{ and}\frac{\beta_{12}^{MB}}{se(\beta_{12}^{MB})}$$
$$=\frac{\beta_{12}}{se(\beta_{12})}\sqrt{\frac{\prod_{i=3}^{m}(1-\beta_{2i}^{2})}{\prod_{i=4}^{m}(1-\beta_{2i}^{2}){+}\sum_{i=4}^{m}[\beta_{2i}^{2}\prod_{j=3,j\neq i}^{m}\left(1-\beta_{2j}^{2}\right)]}}{<}\frac{\beta_{12}}{se(\beta_{12})}.$$

Theorem 3. Consider a general inverse tree with $m + l + 2$ variables, whose structure and parameters are given by $X_i = e_i$, $i = 1,2, \ldots, l, l + 3, \ldots, l + m$, $X_{l+1}{=}\sum_{i=1}^{l}\beta_{i,l+1}X_i{+}e_{l+1}, X_{l+2}{=}\beta_{l+1,l+2}X_{l+1}{+}\sum_{i=3}^{m}\beta_{l+i,l+2}X_{l+i}{+}e_{l+2}$. All the variables have unit variance. If X_{l+1} is regressed on all other variables, i.e., $X_{l+1}{=}\sum_{i=1}^{l}\beta_{i,l+1}^{MB}X_i{+}\beta_{l+1,l+2}^{MB}X_{l+2}{+}\sum_{i=3}^{m}\beta_{l+i,l+2}^{MB}X_{l+i}{+}e_{l+1}^{MB}$, the coefficients for X_{l+1} 's parent, $X_k(k = 1,2, \ldots, l)$, are:

$$\beta_{k,l+1}^{MB}{=}\beta_{k,l+1}\frac{1-\sum_{i=1}^{m}\beta_{l+2+i,l+2}^{2}-\beta_{l+1,l+2}^{2}}{1-\sum_{i=1}^{m}\beta_{l+2+i,l+2}^{2}-\beta_{l+1,l+2}^{2}\sum_{i=1}^{m}\beta_{i,l+1}^{2}}{<}\beta_{k,l+1},\text{ and}\frac{\beta_{k,l+1}^{MB}}{se(\beta_{k,l+1}^{MB})}$$
$$=\frac{\beta_{k,l+1}}{se(\beta_{k,l+1})}\frac{1-\sum_{i=1}^{m}\beta_{l+2+i,l+2}^{2}-\beta_{l+1,l+2}^{2}}{1-\sum_{i=1}^{m}\beta_{l+2+i,l+2}^{2}-\beta_{l+1,l+2}^{2}\sum_{i=1}^{m}\beta_{i,l+1}^{2}}$$
$$\times\sqrt{\frac{(1-\sum_{i=1}^{m}\beta_{i,l+1}^{2})(1-\sum_{i=1}^{m}\beta_{l+2+i,l+2}^{2}-\beta_{l+1,l+2}^{2}\sum_{i=1}^{m}\beta_{i,l+1}^{2})}{(1-\sum_{i=1}^{m}\beta_{l+2+i,l+2}^{2}-\beta_{l+1,l+2}^{2}\sum_{i=1,i\neq k}^{m}\beta_{i,l+1}^{2})}}$$
$$\times\sqrt{\frac{1-\sum_{i=1}^{m}\beta_{i,l+1}^{2}(1-\sum_{i=1}^{m}\beta_{l+2+i,l+2}^{2}-\beta_{l+1,l+2}^{2})}{1-\sum_{i=1,i\neq k}^{m}\beta_{i,l+1}^{2}(1-\sum_{i=1}^{m}\beta_{l+2+i,l+2}^{2}-\beta_{l+1,l+2}^{2})}}{<}\frac{\beta_{12}}{se(\beta_{12})}.$$

For example, consider a general tree with $\beta_{12} = 0.3$, $\beta_{2i} = 0.8$, m = 8. Based on Theorem 2, $\beta_{12}^{MB}{=}0.028\ll\beta_{12}$ and $\beta_{12}^{MB}/se(\beta_{12}^{MB}){=}0.092\sqrt{n-1}\ll\beta_{12}/se(\beta_{12}){=}0.3144\sqrt{n-1}$.

Consider a general inverse tree with seven variables: $X_i, i = 1, \cdots, 5$, are parents of X_6 which is a parent of X_7; $[\beta_{16}, \cdots, \beta_{56}] = [0.24,0.325,0.256,0.304,0.216]$; $\beta_{67} = 0.3$. Based on Theorem 3, $[\beta_{16}^{MB},\cdots,\beta_{56}^{MB}]{=}[0.036,0.053,0.038,0.045,0.032]\ll[\beta_{16},\cdots,\beta_{56}]$ and $[\beta_{16}^{MB}/se(\beta_{16}^{MB}),\cdots,\beta_{56}^{MB}/se(\beta_{56}^{MB})]$ is equal to $[0.1142,0.1672,0.1216,0.1444,0.1026]\times\sqrt{(n-1)}$ which is less than $[\beta_{16}/se(\beta_{16}), \cdots, \beta_{56}/se(\beta_{56})]$ since the later one is equal to

$$[0.306,0.4491,0.3266,0.3878,0.2785]\times\sqrt{(n-1)}.$$

Note that these theorems derive the relationship between β^{MB} ($\beta^{MB}/se(\beta^{MB})$) and β ($\beta/se(\beta)$) at the population-level (i.e., sampling error is not considered), so they use the notation "β" not "$\hat{\beta}$".

5. SIMULATION STUDIES ON SYNTHETIC DATA

We show four sets of simulations. The first one is to show that, on a general tree, the existing algorithms based on the two-stage approach may miss some true parents with high probabilities, while the proposed SBN performs well. The second simulation is to compare the structure learning accuracy of SBN with other competing algorithms, on benchmark networks. The third and fourth simulations are to investigate the scalability of SBN and compare it with other competing algorithms.

5.1 Learning accuracy for general tree

Ten existing algorithms are selected: HITON-PC [25], IAMB and three of its variants [26], GS [17], SC [13], TC and its advanced version TC-bw [18], and L1MB [14]. We simulate data from the general tree in Figure 3(a) with a sample size of 200.

We apply the selected algorithms on the simulated data; the parameters of each algorithm are selected in the way that the authors have suggested in their papers, respectively. In applying the proposed SBN, λ_1 is selected by BIC; λ_2 is set to be $10[(n-1)^2 p/\lambda_1 - \lambda_1]$ which empirically guarantees a DAG to be learned. The initial value of SBN is the output of L1MB which uses LASSO in Stage 1 to identify the MB for each variable. We treat the identified MB by L1MB as parents and use the resulting "BN" (not necessarily a DAG) as the initial value for SBN. The results over 100 repetitions are shown in Figure 3(b). The X-axis records the 10 selected algorithms and the proposed SBN (the last one). The Y-axis records the frequency of a true arc being identified. Each color bar corresponds to a true arc indicated in the legend. Six true arcs are shown, i.e., the arcs between X_1 and X_i, $i = 2, \cdots, 7$. Because X_1 is the parent of X_i, the Y-axis actually shows how well the parent of X_i can be identified by each of the algorithms. Figure 3 (b) shows that, SBN performs much better than all others. This can be explained by Theorem 1. Specifically, the MB of X_i includes X_1 and six children. Although the coefficient linking X_1 to X_i is as high as 0.3, this coefficient in the regression that regresses X_1 on its MB reduces to 0.028, due to the inclusion of the children in the regression. As a result, X_1 will have a high probability of being excluded from the MB identified in Stage 1 of the existing algorithms.

5.2 Learning accuracy for Benchmark networks

We select seven moderately large networks from BNR [27]. We also use the tiling technique to produce two large BNs, alarm2 and hailfinder2. Two other networks with specific structures, factor and chain, are also considered. The 11 networks are: (number of nodes/edges): 1. factors (27/68), 2. alarm (37/46), 3. barley (48/84), 4. carpo (61/74), 5. chain(7/6), 6. hailfinder (56/66), 7. insurance (27/52), 8. mildew (35/46), 9. water (32/66), 10. alarm2 (296/410), 11. hailfinder2 (280/390). To specify the parameters of a network, i.e., to specify the regression coefficients of each variable on its parents, we randomly sample from $\pm 1 + N(0,1/16)$. Then, we simulate data from the networks with sample size 100, and apply the 10 algorithms to learn the BN structures. The results over 100 repetitions are shown in Figure 3(c). The X-axis records the 11 networks. The Y-axis records the ratio of the total learning error (false positives plus false negatives) to the number of arcs in the true BN. Each curve corresponds to one of the 11 algorithms under comparison. We can observe that the lowest curve (i.e., best performance) is SBN.

5.3 Scalability

We study two aspects of scalability for SBN: the scalability with respect to the number of variables in a BN, p, and the scalability with respect to the sample size, n. We use the CPU time for each sweep through all the columns of **B** as the parameter for measurement. Specifically, we fix n =1000, and vary p by using the 11 benchmark networks. Also, we fix p =37 (the Alarm network). The results over 100 repetitions are shown in Figure 4 (a) and (b), respectively. It can be seen that the computational cost is linear in n and quadratic in p, which confirms our theoretical time complexity analysis in Section 3.

6. BRAIN EFFECTIVE CONNECTIVITY MODELING OF AD BY SBN

FDG-PET baseline images of 49 AD and 67 normal control (NC) subjects from the ADNI project (http://www.loni.ucla.edu/ADNI) were used in this study. After spatially normalizing the images to the Montreal Neurological Institute (MNI) template coordinate space, average PET counts were extracted from the 116 brain anatomical regions of interest (AVOIs) defined by the Automated Anatomical Labeling [28] technique. We then selected 42 AVOIs that are considered to be potentially relevant to AD, as reported in the literature. Each of the 42 AVOIs became a node in the SBN. Please see Table 2 for the name of each AVOI brain region. These regions are distributed in the four major neocortical lobes of the brain, i.e., the frontal, parietal, occipital, and temporal lobes.

We apply SBN to learn a BN for AD and another one for NC, to represent their respective effective connectivity models. In the learning of an AD (or NC) effective connectivity model, the value for λ_1 needs to be selected. In this paper, we adopt two criteria in selecting λ_1: one is to minimize the prediction error of the model and the other is to minimize the BIC. Both criteria have been popularly adopted in sparse learning [12,14,15]. The two criteria lead to similar findings discovered from the effective connectivity models, so only the results based on the minimum prediction error are shown in this section. Specifically, Figure 6 shows the effective connectivity models for AD and NC. Each model is represented by a "matrix". Each row/column is one AVOI, X_j. A black cell at the i-th row and j-th column of the matrix represents that X_i is a parent of X_j. On each matrix, four red squares are used to highlight the four lobes, i.e., the frontal, parietal, occipital, and temporal lobes, from top-left to bottom-right. The black cells inside each red square reflect intra-lobe effective connectivity, whereas the black cells outside the squares reflect inter-lobe effective connectivity. The following interesting observations can be drawn from the effective connectivity models:

Global-scale effective connectivity

The total number of arcs in a BN connectivity model, equal to the number of black cells in a matrix plot in Figure 6, represents the amount of effective connectivity (i.e., the amount of directional information flow) in the whole brain. This number is 285 and 329 for AD and

NC, respectively. In other words, AD has 13.4% less effective connectivity than NC. Loss of connectivity in AD has been widely reported in the literature [34, 38–40].

Intra-/inter-lobe effective connectivity distribution

Aside from having different amounts of effective connectivity at the global scale, AD may also have a different distribution pattern of connectivity across the brain from NC. Therefore, we count the number of arcs in each of the four lobes and between each pair of lobes in the AD and NC effective connectivity models. The results are summarized in Table 3. It can be seen that the temporal lobe of AD has 22.9% less effective connectivity than NC. The decrease in connectivity in the temporal lobe of AD has been extensively reported in the literature [29, 2, 30]. The interpretation may be that AD is featured by a dramatic cognitive decline and the temporal lobe is responsible for delivering memory and other cognitive functions. As a result, the temporal lobe is affected early and severely by AD and the connectivity network in this lobe is severely disrupted. On the other hand, the frontal lobe of AD has 27.6% more connectivity than NC. This has been interpreted as compensatory reallocation or recruitment of cognitive resources [29–32]. Because the regions in the frontal lobe are typically affected later in the course of AD (our data are mild to moderate AD), the increased connectivity in the frontal lobe may help preserve some cognitive functions in AD patients. In addition, AD shows a decrease in the amount of effective connectivity in the parietal lobe which has also been reported to be affected by AD. There is no significant difference between AD and NC in the occipital lobe. This is reasonable because the occipital lobe is primarily involved in the brain's visual function which is not affected by AD.

Furthermore, the most significant reduction in inter-lobe connectivity in AD is found between the frontal and temporal lobes, i.e., AD has 29.5% less effective connectivity than NC. This may be attributed to the temporal lobe disruption of the default mode network in AD [34].

Direction of local effective connectivity

One advantage of BNs over undirected graphical models in brain connectivity modeling is that the directed arcs in a BN reflect directional effect of one region over another, i.e., the effective connectivity. Specifically, if there is a directed arc from brain regions X_i to X_j, it indicates that X_i takes a dominant role in thecommunication with X_j. The connectivity modes in Figure 6 reveal a number of interesting findings in this regard:

i. There are substantially fewer black cells in the area defined by rows 27–42 and columns 1–26 in AD than NC. Recall that rows 27–42 correspond to regions in the temporal lobe. Thus, this pattern indicates a substantial reduction in arcs pointing from temporal regions to the other regions in the AD brain, i.e., temporal regions lose their dominating roles in communicating information with the other regions as a result of AD. The loss is most severe in the communication from temporal to frontal regions.

ii. Rows 31 and 35, corresponding to brain regions "Temporal_Mid_L" and "Temporal_Inf_L", respectively, are among the rows with the largest number of black cells in NC, i.e., these two regions take a significantly dominant role in communicating with other regions in normal brains. However, the dominancy of the two regions is substantially reduced by 34.8% and 36.8%, respectively, in AD. A possible interpretation is that these are neocortical regions associated with amyloid deposition and early FDG hypometabolism in AD [34–37, 41–42].

iii. Columns 39 and 40 correspond to regions “Hippocampus_L” and “Hippocampus_R”, respectively. There are a total of 33 black cells in these two columns in NC, i.e., 33 other regions dominantly communicate information with the hippocampus. However, this number reduces to 22 (33.3% reduction) in AD. The reduction is more severe in Hippocampus_L (50%). The hippocampus is well known to play a prominent role in making new memories and in recall. It has been widely reported that the hippocampus is affected early in the course of AD, leading to memory loss – the most common symptom of AD.

iv. There are a total of 93 arcs pointing from the left hemisphere to the right hemisphere of the brain in NC; this number reduces to 71 (23.7% reduction) in AD. The number of arcs from the right hemisphere to the left hemisphere in AD is close to that in NC. This provides evidence that AD may be associated with inter-hemispheric disconnection and the disconnection is mostly unilateral, which has also been reported by some other papers [43–44].

7. CONCLUSION

In this paper, we proposed a BN structure learning algorithm, called SBN, for identifying effective connectivity of AD from FDG-PET data. SBN adopted a novel formulation that involves one L1-norm penalty term to impose sparsity on the learning and another penalty to ensure the learned BN to be a DAG. We performed theoretical analysis on the competitive advantage of SBN over the existing algorithms in terms of learning accuracy. Our analysis showed that the existing algorithms employ a two-stage approach in BN structure identification, which makes them have a high risk of failing to identify the parents of each variable. Also, we performed theoretical analysis on the time complexity of SBN, which showed that it is linear in the sample size and quadratic in the number of variables. Furthermore, we conducted experiments to compare SBN with 10 competing algorithms and found that SBN has significantly better performance in learning accuracy.

We applied SBN to identify the effective connectivity models of AD and NC from PDG-PET data, and found that the effective connectivity of AD is different from NC in many ways. Clinically, our findings may provide an additional tool for monitoring disease progress, evaluating treatment effects, and enabling early detection of network disconnection in prodromal AD. Future studies may be conducted to verify the statistical significance of the identified effective connectivity difference between AD and NC, and estimate the strength of the directed arcs identified by SBN to help improve the sensitivity and specificity of the effective connectivity models. Finally, although this paper focuses on the structure learning of Gaussian BNs, the same formulation can be adopted to discrete BNs, which will be explored in our future research.

Acknowledgments

This research is sponsored in part by NSF IIS-0953662 and NSF MCB-1026710.

Appendix A

There are some notation changes as follows for the convenience of the derivation: we use $b_0 = \beta_{12}$, $b_1 = \beta_{23}$, $b_2 = \beta_{24}$, …, $b_n = \beta_{2m}$, $Y_1 = X_3$, $Y_2 = X_4$, …, $Y_n = X_m$.

Based on Wright's second decomposition rule [33], the covariance matrix of all the variables, T_n, can be represented as a function of the parameters of the BN, $b_0, \cdots, b_n$, i.e.,

$$T_n = cov(X_2, X_1, Y_1, \cdots, Y_n) = \begin{bmatrix} 1 & b_0 & b_1 & b_2 & \cdots & b_n \\ b_0 & 1 & b_0 b_1 & b_0 b_2 & \cdots & b_0 b_n \\ b_1 & b_0 b_1 & 1 & b_1 b_2 & \cdots & b_1 b_n \\ b_2 & b_0 b_2 & b_1 b_2 & 1 & \cdots & b_2 b_n \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ b_n & b_0 b_n & b_1 b_n & b_2 b_n & \cdots & 1 \end{bmatrix} \quad \text{(A-1)}$$

Now, consider the regression of X_2 on all other variables, i.e., $X_2 = b_0^{MB} X_1 + b_1^{MB} Y_1 + \cdots + b_n^{MB} Y_n + e_2^{MB}$. According to the Least Square criterion, the regression coefficients,

$$[b_0^{MB}, \cdots, b_n^{MB}]^T = C_n^{-1} [b_0, \cdots, b_n]^T, \quad \text{(A-2)}$$

where C_n is a sub-matrix of T_n by deleting the 1st column and 1st row from T_n. Denote C_n^{-1} by $A_n = (a_{ij})$Then,

$$b_0^{MB} = a_{11} b_0 + a_{12} b_1 + a_{13} b_2 + \cdots + a_{1n+1} b_n. \quad \text{(A-3)}$$

Our final objective is to express b_0^{MB} by the parameters of the BN. This can be achieved if we can express $a_{11}, a_{12}, a_{13}, \cdots, a_{1n+1}$ by the parameters of the BN, which is the goal of the following derivation.

It is known that

$$a_{1j} = (-1)^{1+j} \frac{\det(C_{ij,n})}{\det(C_n)}, \quad \text{(A-4)}$$

where $C_{1j,n}$ is a matrix by deleting the 1st row and the jth column from C_n. So, the problem becomes calculation of $\det(C_n)$ and det $\det(C_{1j,n})$.

(i) Calculation of $\det(C_n)$: We first show the result:

$$\det(C_n) = \prod_{i=1}^{n} (1 - b_i^2) + \sum_{i=1}^{n} \left(b_i^2 \prod_{j=0, j \neq i}^{n} (1 - b_j^2)\right) \quad \text{(A-5)}$$

Next, we will use the induction method in 1)–2) below to prove (A-5):

1. When $n = 1$, it is easy to see that (A-5) holds.
2. Assume that (A-5) holds for $n - 1$, i.e.,

$$\det(C_{n-1})=\prod_{i=1}^{n-1}(1-b_i^2)+\sum_{i=1}^{n-1}(b_i^2\prod_{j=0,j\neq i}^{n-1}(1-b_j^2)).$$

Then we need prove that (A-5) holds for n. Based on the definition of a determinant,

$$\det(C_n)=\sum_{j=1}^{n+1}(-1)^{1+j}c_{1j}\det(C_{1j,n}),\quad \text{(A-6)}$$

where c_{1j} is the entry at the 1st row and jth column of C_n.

Now we need to derive det($C_{1j,n}$) (only results are shown below due to page limits):

When $j = 1$,

$$\det(C_{1j,n})=\prod_{i=2}^{n}(1-b_i^2)+\sum_{i=2}^{n}(b_i^2\prod_{j=1,j\neq i}^{n}(1-b_j^2)),\quad \text{(A-7)}$$

When $j \neq 1$,

$$\det(C_{1j,n})=(-1)^{1+j+1}b_0b_j\prod_{k=1,k\neq j}^{n}(1-b_k^2).\quad \text{(A-8)}$$

Then, insert (A-7), (A-8), and $c_{11} = 1$, $c_{12} = b_0b_1$, …, $c_{1n+1} = b_0b_n$ into (A-6):

$$\det(C_n)=\prod_{i=2}^{n}(1-b_i^2)+\sum_{i=2}^{n}(b_i^2\prod_{\substack{j\neq i\\ j=1}}^{n}(1-b_j^2))+\sum_{j=1}^{n}-b_0^2b_j^2\prod_{\substack{k=1\\ k\neq j}}^{n}(1-b_k^2)=\prod_{j=1}^{n}(1-b_j^2)+\sum_{j=1}^{n}b_j^2\prod_{\substack{k=0\\ k\neq j}}^{n}(1-b_k^2)$$

This completes the proof for (A-5).

(ii) Calculation of det($C_{1j,n}$): det($C_{1j,n}$)has been obtained by (A-7) and (A-8). Inserting (A-5), (A-7), and (A-8) into (A-4), we get:

$$a_{11}=\frac{\prod_{j=2}^{n}(1-b_j^2)+\sum_{j=2}^{n}(b_j^2\prod_{\substack{k\neq j\\ k=1}}^{n}(1-b_k^2))}{\prod_{i=1}^{n}(1-b_i^2)+\sum_{i=1}^{n}(b_i^2\prod_{\substack{k\neq i\\ k=0}}^{n}(1-b_k^2))}$$

Furthermore, $a_{11} + a_{12}b_0b_1 + a_{13}b_0b_2 + \cdots + a_{1n+1}b_0b_n = 1$. Inserting this into (A-3), we get $b_0^{MB}=a_{11}\left(1-(1-b_0^2)\right)/b_0$. Plugging in the a_{11} above, we can get:

$$b_0^{MB}=b_0 \frac{\prod\limits_{j=1}^{n}(1-b_j^2)}{\prod\limits_{i=1}^{n}(1-b_i^2)+\sum\limits_{i=1}^{n}(b_i^2 \prod\limits_{\substack{k\neq i\\k=0}}^{n}(1-b_k^2))} \quad \text{(A-9)}$$

Obviously, the fraction at the right-hand side is between 0 and 1. Therefore, $|b_0^{MB}|<|b_0|$.

Next we derive the formula for $b_0^{MB}/se(b_0^{MB})$. It is known that $se^2(b_0^{MB})=a_{11}/[(n-1)(T_n^{-1})_{11}]$. Since $(T_n^{-1})_{11}=\det(C_n)/\det(T_n)=\det(C_n)/\prod_{i=0}^{n}(1-b_i^2)$ and det(C_n) is given in (A-5), we can get:

$$se^2(b_0^{MB})=\frac{1}{n-1}\frac{\prod\limits_{i=0}^{n}(1-b_i^2)}{\prod\limits_{i=1}^{n}(1-b_i^2)+\sum\limits_{i=1}^{n}(b_i^2 \prod\limits_{\substack{k\neq i\\k=0}}^{n}(1-b_k^2))}\times\frac{\prod\limits_{j=2}^{n}(1-b_j^2)+\sum\limits_{j=2}^{n}(b_j^2 \prod\limits_{\substack{k\neq j\\k=1}}^{n}(1-b_k^2))}{\prod\limits_{i=1}^{n}(1-b_i^2)+\sum\limits_{i=1}^{n}(b_i^2 \prod\limits_{\substack{k\neq i\\k=0}}^{n}(1-b_k^2))} \quad \text{(A-10)}$$

Also, $se^2(b_0)=(1-b_0^2)/(n-1)se^2(b_0)=(1-b_0^2)/(n-1)$. Putting this together with (A-9) and (A-10), we can get:

$$\frac{b_0^{MB}}{se(b_0^{MB})}=\frac{b_0}{se(b_0)}\times\sqrt{\frac{\prod\limits_{i=1}^{n}(1-b_i^2)}{\prod\limits_{j=2}^{n}(1-b_j^2)+\sum\limits_{j=2}^{n}(b_j^2 \prod\limits_{\substack{k\neq j\\k=1}}^{n}(1-b_k^2))}}$$

It is obvious that the part in the "{ }" is less than one. Therefore,$b_0^{MB}/se(b_0^{MB})<b_0/se(b_0)$.

REFERENCES

1. Horwitz B. The Elusive Concept of Brain Connectivity. Neuroimage. 2003; 19:466–470. [PubMed: 12814595]
2. Azari N, Rapoport S. Patterns of Interregional Correlations of Cerebral Glucose Metabolic Rates in Patients with Dementia of the Alzheimer Type. Neurodegeneration. 1992; 1:101–111.
3. Friston KJ. Functional and Effective Connectivity in Neuroimaging: A Synthesis. Human Brain Mapping. 1994; 2:56–78.
4. Alexander G, Moeller J. Application of the Scaled Subprofile Model to Functional Imaging in Neuropsychiatric Disorders: A Principal Component Approach to Modeling Brain Function in Disease. Human Brain Mapping. 1994; 2:79–94.
5. Calhoun V, Adali T, Pearison G, Pekar J. Spatial and Temporal Independent Component Analysis of Functional MRI Data Containing a Pair of Task-Related Waveforms. HumanBrain Mapping. 2001; 13:43–53.
6. McIntosh A, Bookstein F, Haxby J, Grady C. Spatial Pattern Analysis of Functional Brain Images Using Partial Least Squares. Neuroimage. 1996; 3:143–157. [PubMed: 9345485]

7. Chiang, J.; Wang, Z.; McKeown, MJ. Sparse Multivariate Autoregressive (mAR)-based Partial Directed Coherence (PDC) for EEG Analysis; Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing; 2009. p. 457-460.

8. Huang S, Li J, Sun L, Ye J, Fleisher A, Wu T, Chen K, Reiman E. Learning Brain Connectivity of Alzheimer's Disease by Sparse Inverse Covariance Estimation. NeuroImage. 2010; 50:935–949. [PubMed: 20079441]

9. Hilgetag, C.; Kotter, R.; Stephan, KE.; Sporns, O. Computational Methods for the Analysis of Brain Connectivity. Ascoli, GA., editor. Totowa, NJ: Computational Neuroanatomy. Humana Press; 2002.

10. Bullmore E, Horwitz B, Honey G, Brammer M, Williams S, Sharma T. How Good is Good Enough in Path Analysis of fMRI? Neuroimage. 2000; 11:289–301. [PubMed: 10725185]

11. Friston KJ, Harrison L, Penny W. Dynamic Causal Modeling. Neuroimage. 2003; 19:1273–1302. [PubMed: 12948688]

12. Lam W, Bacchus F. Learning Bayesian Belief Networks: an Approach based on the MDL Principle. Computational Intelligence. 1994; 10:269–293.

13. Friedman N, Nachman I, Pe'er D. Learning Bayesian Network Structure from Massive Datasets: The "Sparse Candidate" Algorithm. UAI 1999. 1999

14. Schmidt M, Niculescu-Mizil A, Murphy K. A. Learning Graphical Model Structures using L1-Regularization Paths. AAAI 2007. 2007

15. Tibshirani R. Regression Shrinkage and Selection via the Lasso. Journal of Royal Statistical Society, Series B. 1996; 58(1):267–288.

16. Tsamardinos I, Brown L, Aliferis C. The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm. Machine Learning. 2006; 65(1):31–78.

17. Margaritis D, Thrun S. Bayesian Network induction via local neighborhoods. NIPS 1999. 1999

18. Pellet J, Elisseeff A. Using Markov Blankets for Causal Structure Learning. JMLR. 2008; 9:1295–1342.

19. Meek C. Causal inference and causal explanation with background knowledge. UAI 1995. 1995

20. Estrada E, Naomichi H. Communicability in Complex Networks. Phys. Rev. E. 2008; 77 036111.

21. Gill, PE. Class Notes for Math 271ABC: Numerical Optimization. UCSD: Department of Mathematics;

22. Bertsekas, D. Nonlinear Programming. 2nd Edition. Belmont: Athena Scientific; 1999.

23. Fu W. Penalized Regressions: the Bridge versus the Lasso. Computational and Graphical Statistics. 1998; 7(3):397–416.

24. Friedman J, Hastie T, Hofling H, Tibshirani R. Pathwise Coordinate Optimization. The Annals of Applied Statistics. 2007; 2:302–332.

25. Aliferis C, Tsamardinos I, Statnikov A. HITON, a Novel Markov Blanket Algorithm for Optimal Variable Selection. AMIA 2003. 2003:21–25.

26. Tsamardinos I, Aliferis C. Towards Principled Feature Selection: Relevancy, Filters and Wrappers. AISTAT 2003. 2003

27. http://www.cs.huji.ac.il/labs/compbio/Repository.

28. Tzourio-Mazoyer N, Landeau B, Papathanassiou D, Crivello F, Etard O, Delcroix N, Mazoyer B, Joliot M. Automated Anatomical Labeling of Activations in SPM using a Macroscopic Anatomical Parcellation of the MNI MRI Single-Subject Brain. Neuroimage. 2002; 15:273–289. [PubMed: 11771995]

29. Supekar K, Menon V, Rubin D, Musen M, Grecius M. Network Analysis of Intrinsic Functional Brain Connectivity in AD. PLoS Comput Biol. 2008; 4(6):1–11.

30. Wang K, Liang M, Wang L, Tian L, Zhang X, Li K, Jiang T. Altered Functional Connectivity in Early AD: A Resting-State fMRI Study. Human Brain Mapping. 2007; 28:967–978. [PubMed: 17133390]

31. Gould R, Arroyo B, Brown R, Owen A, Bullmore E, Howard R. Brain Mechanisms of Successful Compensation during Learning in AD. Neurology. 2006; 67(6):1011–1017. [PubMed: 17000970]

32. Stern Y. Cognitive Reserve and Alzheimer Disease. Alzheimer Disease Associated Disorder. 2006; 20:69–74.

33. Korb, KB.; Nicholson, AE. Bayesian Artificial Intelligence. London, UK: Chapman & Hall/CRC; 2003.

34. Greicius M, Srivastava G, Reiss A, Menon V. Default-mode Network Activity Distinguishes AD from Healthy Aging: Evidence from Functional MRI. PNAS. 2004; 101:4637–4642. [PubMed: 15070770]

35. Alexander GE, Chen K, Pietrini P, Rapoport S, Reiman E. Longitudinal PET Evaluation of Cerebral Metabolic Decline in Dementia: A Potential Outcome Measure in AD Treatment Studies. Am.J.Psychiatry. 2002; 159:738–745. [PubMed: 11986126]

36. Braak H, Braak E. Evolution of the Neuropathology of Alzheimer's Disease. Acta Neurol Scand Suppl. 1996; 165:3–12. [PubMed: 8740983]

37. Braak H, Braak E, Bohl J. Staging of Alzheimer-Related Cortical Destruction. Eur Neurol. 1993; 33:403–408. [PubMed: 8307060]

38. Hedden T, Van Dijk K, Becker J, Mehta A, Sperling R, Johnson K, Buckner R. Disruption of Functional Connectivity in Clinically Normal Older Adults Harboring Amyloid Burden. J. Neurosci. 2009; 29:12686–12694. [PubMed: 19812343]

39. Andrews-Hanna J, Snyder A, Vincent J, Lustig C, Head D, Raichle M, Buckner R. Disruption of Large-Scale Brain Systems in advanced Aging. Neuron. 2007; 56:924–935. [PubMed: 18054866]

40. Wu X, Li R, Fleisher AS, Reiman E, Guan X, Zhang Y, Chen K, Yao L. Altered Default Mode Network Connectivity in Alzheimer’s Disease - A Resting Functional MRI and BN Study. Human Brain Mapping. 2011 in press.

41. Ikonomovic M, Klunk W, Abrahamson E, Mathis C, Price J, Tsopelas N, Lopresti B, et al. Post-mortem Correlates of in vivo PiB-PET Amyloid Imaging in a Typical Case of Alzheimer's Disease. Brain. 2008; 131:1630–1645. [PubMed: 18339640]

42. Klunk W, Engler H, Nordberg A, et al. Imaging brain amyloid in Alzheimer's disease with Pittsburgh Compound-B. Ann Neurol. 2004; 55:306–319. [PubMed: 14991808]

43. Reuter-Lorenz P, Mikels J. A Split-Brain Model of Alzheimer’s Disease? Behavioral Evidence for Comparable intra and interhemispheric Decline. Neuropscyhologia. 2005; 43:1307–1317.

44. Lipton A, Benavides R, Hynan LS, Bonte FJ, Harris TS, White CL, Bigio EG 3rd. Lateralization on Neuroimaging does not Differentiate Frontotemporal Lobar Degeneration from Alzheimer’s Disease. Dement Geriatr Cogn Disord. 2004; 17(4):324–327. [PubMed: 15178946]

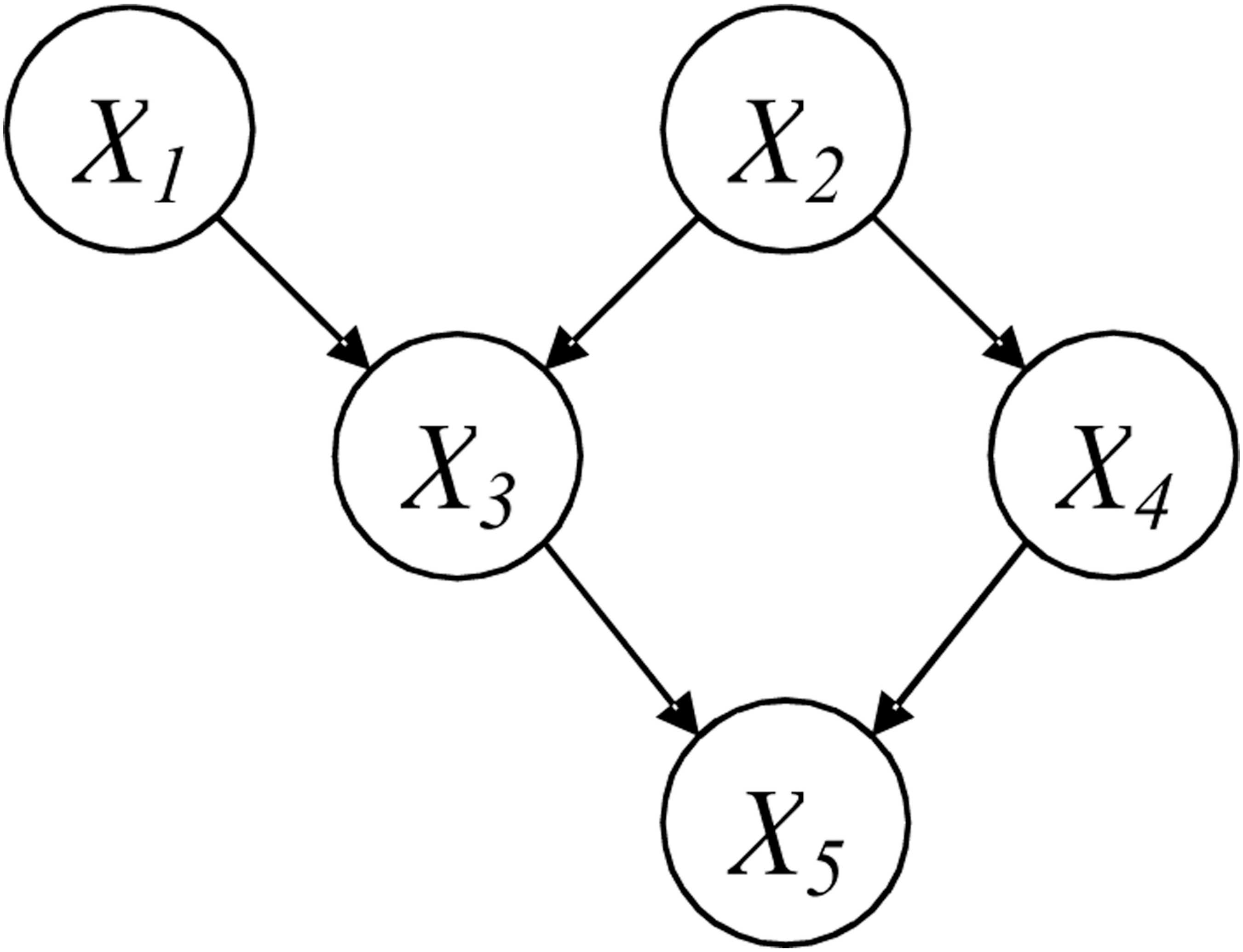
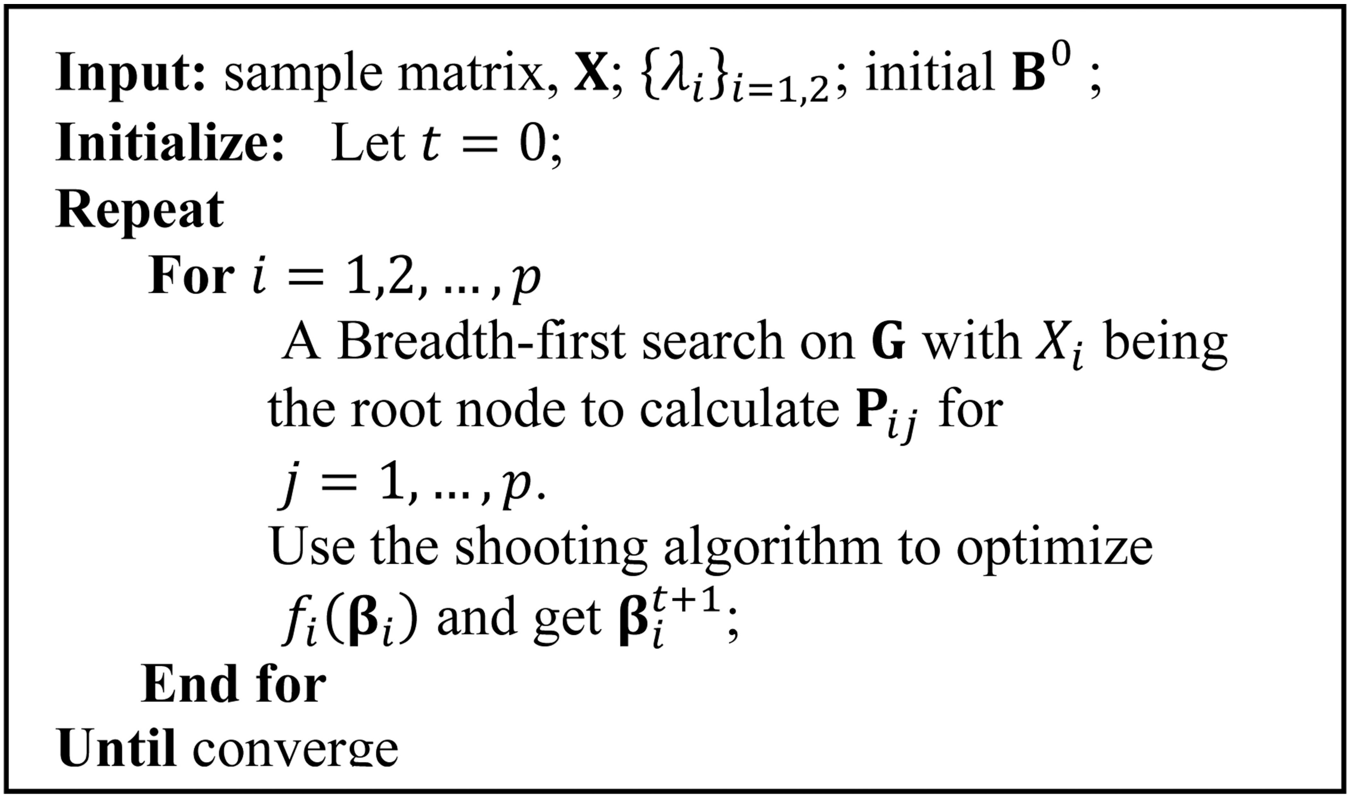


Figure 1.
A BN structure



```
Input: sample matrix, $\mathbf{X}$; $\{\lambda_i\}_{i=1,2}$; initial $\mathbf{B}^0$ ;
Initialize:   Let $t = 0$;
Repeat
    For $i = 1,2, \dots, p$
        A Breadth-first search on $\mathbf{G}$ with $X_i$ being
        the root node to calculate $\mathbf{P}_{ij}$ for
        $j = 1, \dots, p$.
        Use the shooting algorithm to optimize
        $f_i(\boldsymbol{\beta}_i)$ and get $\boldsymbol{\beta}_i^{t+1}$;
    End for
Until converge
```

Figure 2.

The BCD algorithm used for solving (2)

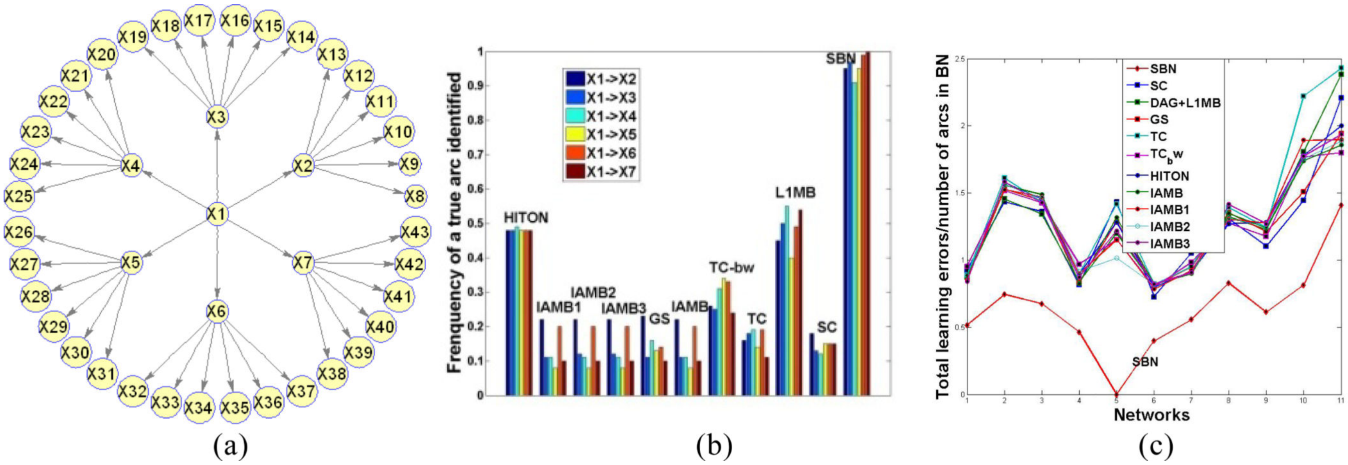


Figure 3.

(a): The general tree in Sec. 5.1 (regression coefficients are 0.3 for arcs between X_1 and X_i, $i = 2, \dots, 7$ and 0.8 for others); (b) Simulation results for the general tree; (c) Simulation results for 11 benchmark networks.

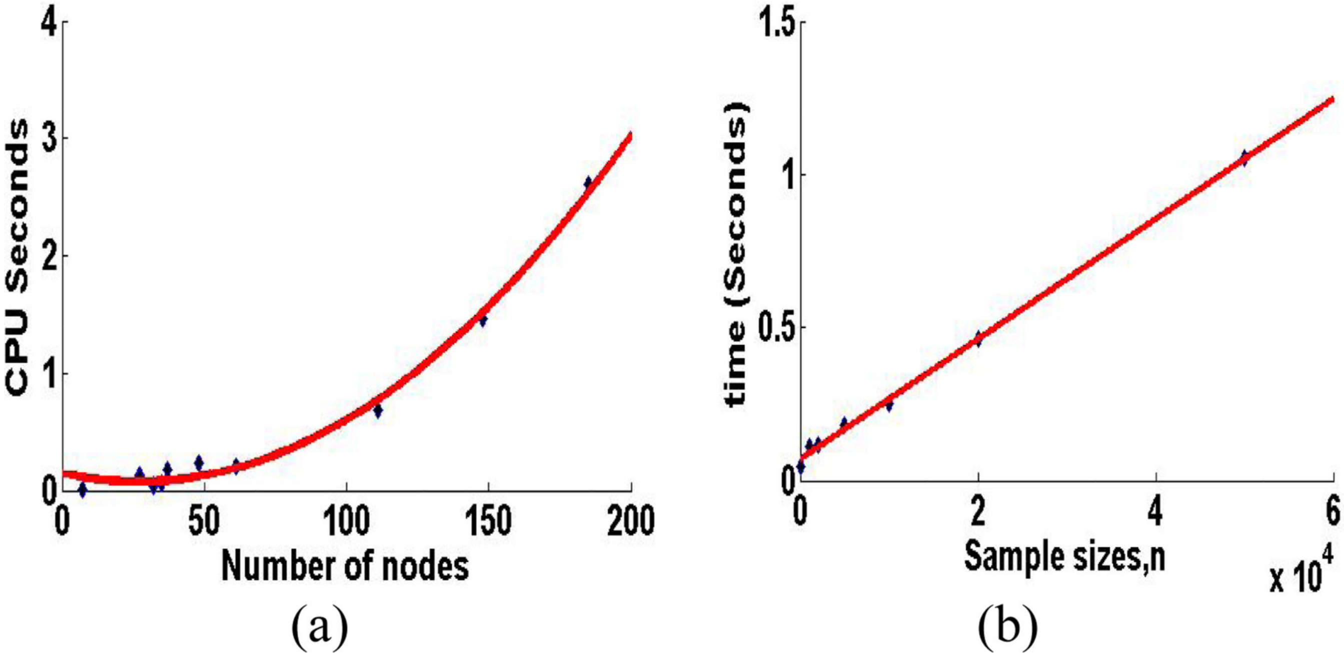


Figure 4.

Scalability of SBN with respect to (a) the number of variables in a BN, p, and (b) the sample size, n, on a computer with Intel Core 2, 2.2 G Hz, 4G

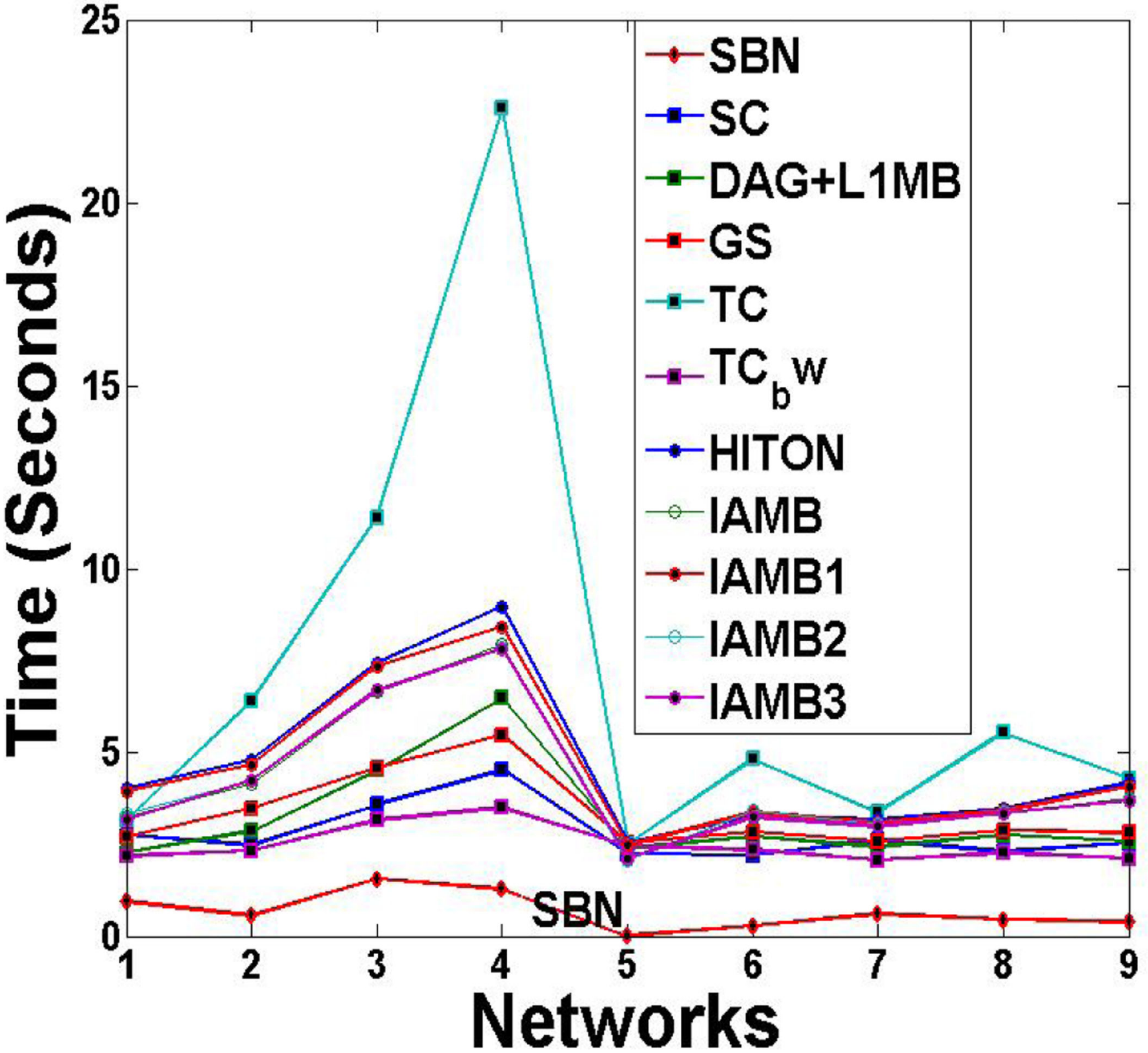


Figure 5.

Comparison of SBN with competing algorithms on the CPU time in structure learning of the other nine benchmark networks

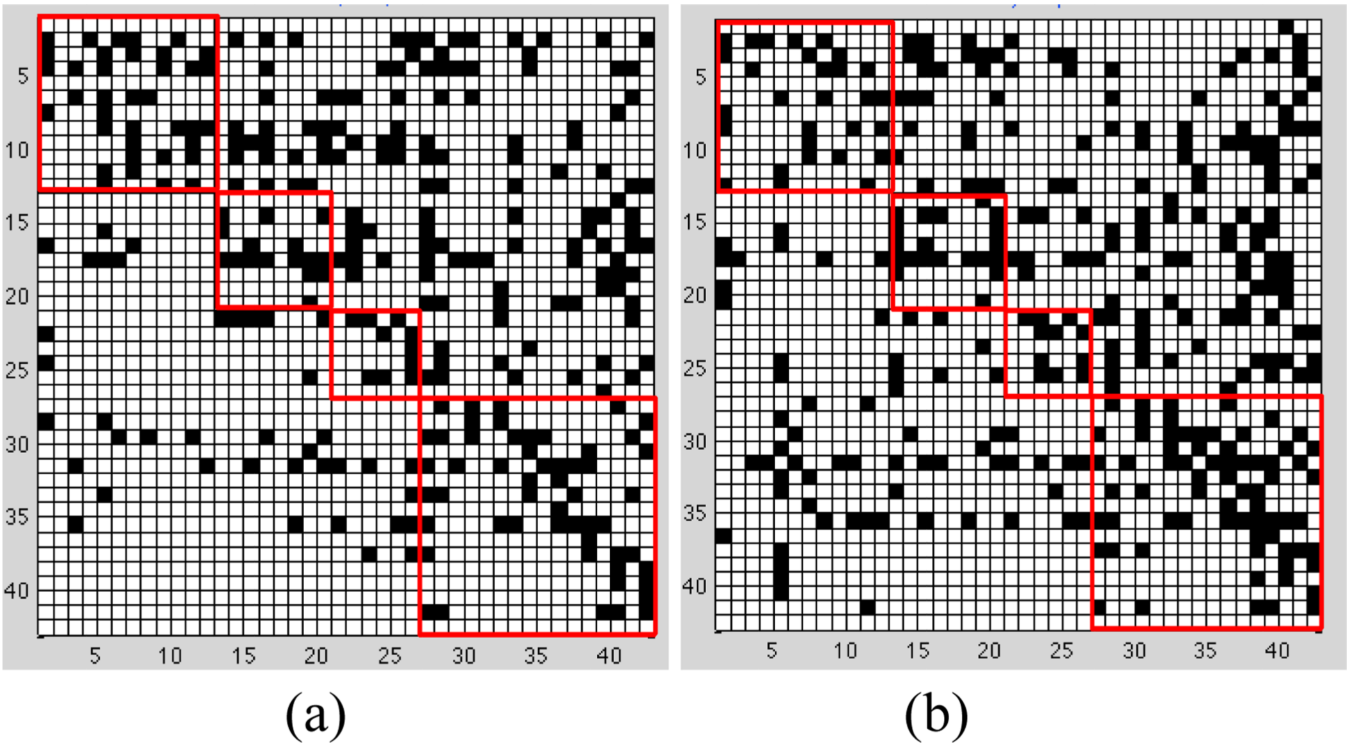


Figure 6.

Brain effective connectivity models by SBN. (a) AD. (b) NC.

Table 2

Names of the AVOIs for effective connectivity modeling (L = left hemisphere, R = right hemisphere)

Frontal lobe		Parietal lobe		Occipital lobe		Temporal lobe	
1	Frontal_Sup_L	13	Parietal_Sup_L	21	Occipital_Sup_L	27	Temporal_Sup_L
2	Frontal_Sup_R	14	Parietal_Sup_R	22	Occipital_Sup_R	28	Temporal_Sup_R
3	Frontal_Mid_L	15	Parietal_Inf_L	23	Occipital_Mid_L	29	Temporal_Pole_Sup_L
4	Frontal_Mid_R	16	Parietal_Inf_R	24	Occipital_Mid_R	30	Temporal_Pole_Sup_R
5	Frontal_Sup_Medial_L	17	Precuneus_L	25	Occipital_Inf_L	31	Temporal_Mid_L
6	Frontal_Sup_Medial_R	18	Precuneus_R	26	Occipital_Inf_R	32	Temporal_Mid_R
7	Frontal_Mid_Orb_L	19	Cingulum_Post_L			33	Temporal_Pole_Mid_L
8	Frontal_Mid_Orb_R	20	Cingulum_Post_R			34	Temporal_Pole_Mid_R
9	Rectus_L					35	Temporal_Inf_L 8301
10	Rectus_R					36	Temporal_Inf_R 8302
11	Cingulum_Ant_L					37	Fusiform_L
12	Cingulum_Ant_R					38	Fusiform_R
						39	Hippocampus_L
						40	Hippocampus_R
						41	ParaHippocampal_L
						42	ParaHippocampal_R

Table 3

Intra – and inter – lobe effective connectivity amounts

(a) AD

	Frontal	**Parietal**	**Occipital**	**Temporal**
Frontal	37	28	18	43
Parietal		16	14	42
Occipital			10	23
Temporal				54

(b) NC

	Frontal	**Parietal**	**Occipital**	**Temporal**
Frontal	29	32	12	61
Parietal		20	16	42
Occipital			11	36
Temporal				70