



HHS Public Access

Author manuscript

IEEE Trans Pattern Anal Mach Intell. Author manuscript; available in PMC 2017 November 30.

Published in final edited form as:

IEEE Trans Pattern Anal Mach Intell. 2016 November ; 38(11): 2269–2283. doi:10.1109/TPAMI.2015.2511754.

Learning Discriminative Bayesian Networks from High-dimensional Continuous Neuroimaging Data

Luping Zhou,

School of Computing and Information Technology, University of Wollongong, NSW 2500, Australia

Lei Wang,

School of Computing and Information Technology, University of Wollongong, NSW 2500, Australia

Lingqiao Liu,

School of Computer Science, University of Adelaide, Australia

Philip Ogunbona, and

School of Computing and Information Technology, University of Wollongong, NSW 2500, Australia

Dinggang Shen

Department of Radiology, University of North Carolina at Chapel Hill, USA

Abstract

Due to its causal semantics, Bayesian networks (BN) have been widely employed to discover the underlying data relationship in exploratory studies, such as brain research. Despite its success in modeling the probability distribution of variables, BN is naturally a generative model, which is not necessarily discriminative. This may cause the ignorance of subtle but critical network changes that are of investigation values across populations. In this paper, we propose to improve the discriminative power of BN models for continuous variables from two different perspectives. This brings two general discriminative learning frameworks for Gaussian Bayesian networks (GBN). In the first framework, we employ Fisher kernel to bridge the generative models of GBN and the discriminative classifiers of SVMs, and convert the GBN parameter learning to Fisher kernel learning via minimizing a generalization error bound of SVMs. In the second framework, we employ the max-margin criterion and build it directly upon GBN models to explicitly optimize the classification performance of the GBNs. The advantages and disadvantages of the two frameworks are discussed and experimentally compared. Both of them demonstrate strong power in learning discriminative parameters of GBNs for neuroimaging based brain network analysis, as well as maintaining reasonable representation capacity. The contributions of this paper also include a new Directed Acyclic Graph (DAG) constraint with theoretical guarantee to ensure the graph validity of GBN.

Index Terms

Bayesian network; discriminative learning; Fisher kernel learning; max-margin; brain network

1 Introduction

As an important probabilistic graphical model, Bayesian network (BN) has been used to model the probability distribution of a set of random variables for a wide spectrum of applications, e.g., diagnosis, troubleshooting, web mining, meteorology and bioinformatics. It combines graph representation with Bayesian analysis, providing an effective way to model and infer the conditional dependency of the variables. A BN has to be a directed acyclic graph (DAG). Two factors characterize a BN, i.e., the structure of the network (the presence/absence of edges in the graph) and the parameters of the probability distribution. Recent research of BN focuses on how to learn the structure and the parameters of BN directly from the data.

The approaches of learning BN structures can be roughly categorized into the constraint-based, the score-based, and the hybrid approaches. The constraint-based approaches use a series of conditional independence testing to ensure the model structure is consistent with the conditional independency entailed by the observations. Methods in this class include the IC algorithm [1], PC algorithm [2], and more recent methods [3], [4]. Score-based approaches define a scoring function over the space of candidate DAGs and optimize this function through certain search strategies. Methods in this class vary with scoring criteria, e.g., the posterior probability [5], [6], [7] and the minimum description length [8], or vary with search strategies, e.g., the heuristic search [9] and the Monte Carlo methods [5]. Hybrid approaches usually employ constraint-based methods to prune the search space of DAG structures and consequently restrict a subsequent score-based search [10], [11]. Many existing BN learning methods, such as LIMB-DAG [12], MMHC [10], TC and TC-bw [13], comprise of two stages: the identification of candidate parent sets in the first stage and the further pruning of them based on certain criteria in the second stage. Despite the mitigation of computational complexity, a drawback arises that if a parent node is missed in the first stage, it will never be recovered in the second stage [14]. To address this issue, one-stage learning process has been preferred in recent research work [14], [15]. In these studies, based on Gaussian Bayesian network (GBN), the parent sets of all variables are learned together to optimize a LASSO-based score function in a single stage. The related optimization problems are solved either approximately [14] or exactly [15]. They have demonstrated improved reliability of BN edge identification over traditional two stage methods.

Although BN is naturally a generative method, it has also been used in classification tasks for diagnostic or predictive purposes. A straightforward usage is to train each class a BN and classify a new sample into the class with the highest likelihood value [14]. Another kind of approaches trains “Bayesian network classifiers” with discriminative objective functions [16], [17], [18]. In these approaches, usually a single BN is learned to optimize the discrimination performance. Either the structure or the parameters of the BN are adjusted to reflect the class difference for better classification. Therefore, the resulting BN does not model the distribution of any individual class. The “Bayesian network classifiers” in [16], [17], [18] are designed for discrete variables of multinomial distribution. They still inherit the two-stage learning process, i.e., have to predefine candidate parent sets as mentioned above.

Learning BN from the data faces new challenges in exploratory domains, such as brain research, where the mechanism of brain and mental diseases remain unclear and need to be explored. These domains usually cater for both interpretation and discrimination. “Interpretation” requires interpretable models of the data and the findings explained by domain language rather than mathematical terms. This requirement comes from the demand of understanding the domain problems. “Discrimination” requires the models to have sufficient discriminative power to distinguish groups of interest (such as identifying the diseased from the healthy), for the purpose of prediction. To some extent, a high accuracy of the predictive model also provides a measure of the amount of information captured by that model.

Being a generative method, BN represents the distribution of the data and is naturally amenable for interpretation. However, it is known that generative methods are not necessarily discriminative. They are prone to emphasizing major structures that are shared within each group, and neglecting the subtle but critical changes across groups. The latter, unfortunately, often happens, for example, in disease-induced brain changes across clinical groups. Consequently, generative methods are usually inferior in prediction compared with the discriminative methods that target only the boundary of classes (such as Support Vector Machines (SVMs)). On the other hand, discriminative methods often encounter the difficulty of interpretation, which is critical in exploratory research aimed at both the understanding and the prediction. Thus, this paper is motivated by the advantages that can be gained by learning BNs that are both representative and discriminative. Different from the Bayesian network classifiers in [16], [17], [18] that address discrete variables, we learn discriminative BNs for continuous variables, which is often needed in many domains including neuroimaging-based brain research. Moreover, we learn for each class a BN with enhanced discrimination and maintain the BN representation of each individual class for interpretation¹. To achieve our goal, we propose two discriminative learning frameworks based on sparse Gaussian Bayesian network (SGBN).

In the first framework (termed KL-SGBN), we employ Fisher kernel [19] to link the generative models of SGBN to the discriminative classifiers of SVMs, and convert the SGBN parameter learning to Fisher kernel learning via maximizing a generalization bound of SVMs. The contributions of this framework include the following. i) By inducing Fisher kernel on SGBN models, we provide a way to obtain sample-specific SGBN-induced feature vectors that can be used by the discriminative classifiers such as SVMs. Through this, we bridge the generative models and the discriminative classifiers. ii) We propose a kernel learning approach to discriminatively learn the parameters of SGBNs by optimizing the performance of SVM. iii) As a by-product, the manipulation of Fisher kernel on SGBN provides a new way of variable selection for SGBNs. This framework has a computational advantage: through the mapping of Fisher kernel, the SGBN-induced feature vectors become

¹In this paper, we deal with the scenario that maintaining the BN representation of individual class is critical for the understanding of domain problems, such as the brain network models for the healthy and the diseased groups. However, it is not difficult to see our discriminative learning frameworks could be slightly modified to learn only a single BN as the existing “Bayesian network classifiers” for continuous variables. However, this deviates from our motivation and therefore is not unfolded in this paper.

linear functions of the SGBN parameters, which significantly simplifies the optimization problem in the learning process.

Unlike KL-SGBN where the discrimination is obtained by optimizing the classification performance of SVMs, in the second learning framework (termed MM-SGBN), we propose to optimize a criterion directly built upon the classification performance of SGBNs. The motivation is that optimizing the performance of SVMs may not necessarily guarantee an equivalent improvement on SGBNs when SGBNs are the goal of applications. The contribution of this framework is a max-margin based method to jointly learn SGBNs, one for each class, for both representation and discrimination.

In addition to the two discriminative SGBN learning frameworks, our contributions in this paper also include a new DAG constraint of SGBN based on topological ordering to ensure the validity of the graph. This new DAG constraint circumvents the awkward hard binarization of SGBN parameters in the process of optimization in [14], and simplifies the related optimization problems. This consequently makes it possible to optimize all the SGBN parameters together to avoid the influence of feature ordering encountered in the Block Coordinate Descent (BCD) optimization in [14]. Moreover, this new DAG constraint also circumvents the need for presetting candidate parent sets as in [17].

Although the discriminative learning frameworks proposed in this paper are general methods, we focus on their applications in neuroimaging analysis for the early diagnosis of mental diseases. A newly emerging field in the imaging-based neuroscience, called brain network analysis, attempts to model the brain as a complex network and study the interactions of brain regions via imaging-based features [20]. Such research is important because brain network change is often found to be a response of the brain to damages. Due to its causal semantics, BN has been employed to model the “effective connectivity” of the brain [14], [21], [22]. The directionality of the connections may disclose the pathways of how one brain region affects another. The discoveries may lend further credence to the evidence of causal relationship changes found in many mental diseases, such as the Alzheimer’s disease (AD) [23], [14], [24], [22], and uncover novel connectivity-based biomarkers for disease diagnosis. The proposed learning frameworks has been tested on multiple neuroimaging data sets. As demonstrated, our methods can significantly improve the discriminative power of the obtained SGBNs, as well as maintaining their representation capacity.

Early conference versions of this work were published in [25], [26]. In this paper, a significant extension has been made on the following aspects. First, we analyze the problems of the DAG constraint used in [25], [26], [14], and propose a new constraint with theoretically guaranteed DAG property to overcome those drawbacks. Second, we experimentally verify the new DAG constraint on benchmark Bayesian network data sets for network structure learning, and compare our method with another eight competing methods in the literature. Third, we update our two discriminative learning frameworks with the new DAG constraint and redo all the experiments in our early work [25], [26]. Fourth, we analyze the connections and differences between the two proposed discriminative learning

frameworks, and conduct more comprehensive experiments to explore the characteristics of our frameworks with varied parameters, which has not been done in [25], [26].

The rest of the paper is organized as follows. Section 2 reviews SGBN and introduces the background of brain network analysis. Sections 3 elaborates two frameworks to learn discriminative and representative SGBNs from continuous data. Section 4 revisits the problem of the existing DAG constraint of SGBN, and proposes a new one based on topological ordering. The proposed two learning frameworks with the new DAG constraint are experimentally tested in Section 5. This paper is concluded in Section 6. The notations of symbols frequently occurring in this paper are summarized in Table 1.

2 Background

To make this paper self-contained, we introduce the background for both the methodology and its application to brain network analysis. Please note that the methodology could be generalized to applications beyond the example given in this paper.

2.1 Sparse Gaussian Bayesian Network (SGBN)

Because this paper is based on SGBN model, in the following, we review the fundamentals of SGBN in [14]. All the symbols are defined in Table 1.

A Bayesian network (BN) \mathcal{G} is a directed acyclic graph (DAG), i.e. there is no closed path within the graph. It expresses the factorization property of a joint distribution

$p(\mathbf{x}) = \prod_{i=1, \dots, m} p(x_i | \mathbf{Pa}_i)$. The conditional probability $p(x_j | \mathbf{Pa}_j)$ is assumed to follow a Gaussian distribution in Gaussian Bayesian Network (GBN). Each node x_j is regressed over its parent nodes $\mathbf{Pa}_j: x_j = \boldsymbol{\theta}_j^\top \mathbf{Pa}_j + \varepsilon_j$, where the vector $\boldsymbol{\theta}_j$ is the regression coefficients, and $\varepsilon_j \sim \mathcal{N}(0, \sigma_j^2)$. The structure of BN could be characterized by the $m \times m$ matrix \mathbf{G} or \mathbf{P} (defined in Table 1), representing the *edges/paths* in the graph, respectively.

Identifying parent sets is critical for BN learning. Traditional methods often consist of two stages: the candidate parent sets are initially identified in the first stage and further pruned by some criteria in the second stage. A drawback arises that when a true parent is missing in the first stage, it will never be recovered in the second stage. The work in [14] proposed a different approach based on sparse GBN (SGBN), denoted as H-SGBN in this paper. In H-SGBN, each node x_j is regressed over all the other nodes, and its parent set is implicitly selected by the regression coefficients $\boldsymbol{\theta}_j$ that are estimated through a constrained LASSO regression. The following optimization is solved in [14]:

$$\begin{aligned} \min_{\boldsymbol{\Theta}} \sum_{i=1}^m \|\mathbf{x}_{:,i} - \mathbf{P} \boldsymbol{\theta}_i\|_2^2 + \lambda_1 \|\boldsymbol{\theta}_i\|_1 \\ \text{s.t. } \boldsymbol{\Theta}_{ji} \times \mathbf{P}_{ij} = 0, \forall i, j = 1, \dots, m, \quad i \neq j. \end{aligned} \quad (2.1)$$

A challenge for BN learning is how to enforce the DAG property, i.e., avoiding directed cycles in the graph. A sufficient and necessary condition for being a DAG is proposed in

[14], which requires $\Theta_{ji} \times \mathbf{P}_{ij} = 0$ for all i and j . Note that \mathbf{P}_{ij} is an implicit function of Θ_{ji} (i.e., $\mathbf{P} = \exp(\Theta)$, the matrix exponential function of Θ , as in [14]). Eqn. (2.1) is difficult to solve. In [14], a block coordinate descent (BCD) method is employed to solve a LASSO-like problem efficiently. The whole Θ is optimized column-wisely and iteratively. In each iteration t , only one column of Θ , say $\Theta_{:,j}$, is optimized with \mathbf{P} fixed as $\mathbf{P}^{(t-1)}$ in the last iteration. Then $\Theta^{(t)}$, with the updated column $\Theta_{:,j}$, is binarized to obtain $\mathbf{G}^{(t)}$, based on which, $\mathbf{P}^{(t)}$ is recalculated by a Breadth-first search with x_j being the root node. The process is repeated until convergence. H-SGBN simultaneously obtains the structure and the parameters of an SGBN via learning Θ , e.g., there is no edge $i \rightarrow j$ if Θ_{ij} is zero. It has been demonstrated to outperform the conventional two-stage methods in network edge recovery.

2.2 Brain Network Analysis

Neuroimaging modalities and analysis techniques can provide more sensitive and consistent measurements than traditional cognitive assessment for the early diagnosis of disease. Many mental disorders are found associated with subtle abnormalities distributed over the entire brain, rather than an individual brain region. The “distributive” nature of mental disorders suggests the alteration of interactions between brain regions (neuronal systems) and thus the necessity of studying the brain as a complex network. Brain networks are mathematically represented by graphical models, which can be constructed from neuroimaging data as follows. The brain images belonging to different subjects are first spatially aligned to a common stereotaxic space by affine or deformable transformation, and then partitioned into regions of interest (ROI), i.e., clusters of imaging voxels, using either data-driven methods or predefined brain atlas. A brain network is then modeled by a graph with each node corresponding to a brain region and each edge corresponding to the connectivity between regions. Brain network analysis studies three kinds of brain connectivity. In this paper, we focus on the “effective connectivity” that describes the influence one brain region exert upon another. Some early works in this field require a prior model of brain connectivity and most have only considered a small number (< 10) of brain regions using techniques such as structural equation modeling [27] and dynamic causal modeling [28]. More recently, models such as BN and Granger Causality have also been introduced into this field. It is suggested that BN may have advantages over those lag-based methods for brain network analysis by an experimental fMRI study [21]. Among BN-related methods, it is worth noting that the work in [14] is completely data-driven, which recovers SGBN from more than 40 brain regions in fluorodeoxyglucose PET (FDG-PET) images. The method employs the strategy of sparsity constraint to handle relatively larger scale BN construction, and circumvents the traditional two-stage procedure for identifying parent sets in many sparse BN learning methods [12], [10].

3 Proposed Discriminative Learning of Generative SGBN

BN models are by definition generative models, focusing on how the data could be generated through an underlying process. In the context of neuroimage analysis, these models represent the effective brain connectivity of the given population. When used for classification, e.g., identifying AD patients from the healthy, the SGBN models are trained for each class separately. A new sample \mathbf{x}_i is then assigned to the class with the higher

likelihood of SGBN. This may ignore some subtle but critical network differences that distinguish the classes. Therefore, we argue that the parameters of the generative model should be learned from the two classes jointly to keep the essential discrimination.

Integrating generative and discriminative models is an important research topic in machine learning. In [29], the related approaches are roughly divided into three categories: blending, staging and iterative methods. In blending methods, both the discriminative and the generative terms are incorporated into the same objective function. In staging methods, the discriminative model is trained on features provided by the generative model. In iterative methods, the generative and the discriminative models are trained iteratively to influence each other. In this paper, we propose two kinds of discriminative learning frameworks to achieve our goal. One is a staging method, called Fisher-kernel-induced discriminative learning (KL-SGBN). It extracts sample-based features from SGBN by Fisher kernel to optimize the classification performance of SVM. The other is a blending method, called max-margin-based discriminative learning (MMSGBN). It directly optimizes the classification performance of SGBNs subject to maintaining SGBN's representation capacity. The two frameworks are elaborated in the following sections, respectively.

3.1 Proposed Fisher-kernel-induced Discriminative Learning (KL-SGBN)

We first introduce the Fisher-kernel-induced discriminative learning of SGBN, i.e., KL-SGBN. The algorithm is illustrated in Fig. 1 and overviewed as follows. Given two classes in comparison, two SGBN models (with the parameters of Θ_1 and Θ_2) are learned, one for each individual class. The original samples are then mapped into the gradient space of the SGBN parameters Θ_1 and Θ_2 by Fisher kernel (Section 3.1.1). Through this mapping, each sample is represented by a new feature vector (called Fisher vector [19]) that is a function of $\Theta = [\Theta_1, \Theta_2]$. These sample-specific feature vectors are then fed into an SVM classifier to minimize its generalization errors by adjusting Θ (Section 3.1.2). The obtained optimal Θ_1^* and Θ_2^* encode the discriminative information and therefore improve the original SGBNs. In this way, we convert the discriminative learning of SGBN parameters to the discriminative learning of Fisher kernels.

3.1.1 Induction of Fisher vectors from SGBN—Below we introduce how to use Fisher kernel on SGBNs to obtain feature vectors required for kernel learning.

Fisher kernel [19] provides a way to compare samples induced by a generative model. It maps a sample to a feature vector in the gradient space of the model parameters. The intuition is that similar objects induce similar log-likelihood gradients of the model parameters. Fisher kernel is computed as $K(\mathbf{x}, \mathbf{x}') = \mathbf{g}_{\mathbf{x}}^T \mathbf{U}^{-1} \mathbf{g}_{\mathbf{x}'}$, where the Fisher vector $\mathbf{g}_{\mathbf{x}} = \nabla_{\theta} \log(p(\mathbf{x}|\theta))$ describes the changing direction of parameters to better fit the model. The Fisher information metric \mathbf{U} weights the similarity measure, but is often set as an identity matrix in practice [19].

Fisher kernel has recently witnessed successful applications in image categorization [30], [31] for inducing feature vectors from Gaussian Mixture Model (GMM) of a visual vocabulary. Despite its success, in the applications above, Fisher kernel is mainly used as a

feature extractor². It has not been applied to learning the parameters of probability distributions before the early work of this paper in [25]. The advantage of learning discriminative Fisher kernel has also been confirmed by a recent study that maximizes the class separability [33] of samples based on Fisher kernel, which is developed with different context and different criteria from ours.

Following [14], we only consider Θ as parameters and predefine σ . Let $\mathcal{L}(\mathbf{x}|\Theta) = \log(p(\mathbf{x}|\Theta))$ denote the log-likelihood. Our Fisher vector for each sample \mathbf{x} is

$$\Phi_{\Theta}(\mathbf{x}) = [\nabla_{\Theta_1} \mathcal{L}(\mathbf{x}|\Theta_1)^\top, \nabla_{\Theta_2} \mathcal{L}(\mathbf{x}|\Theta_2)^\top]^\top,$$

where Θ_1 and Θ_2 are the parameters of the SGBNs for the two classes ($y = 1, 2$), respectively. Recall that, using a BN, the probability $p(\mathbf{x}|\Theta)$ can be factorized as

$$p(\mathbf{x}|\Theta) = \prod_{i=1, \dots, m} p(x_i | \mathbf{P}\mathbf{a}_i, \boldsymbol{\theta}_i). \quad \text{Therefore, for GBN it can be shown that}$$

$$\begin{aligned} \mathcal{L}(\mathbf{x}|\Theta) &= \sum_{i=1}^m \log p(x_i | \mathbf{P}\mathbf{a}_i, \boldsymbol{\theta}_i) \\ &= \sum_{i=1}^m \left[\frac{-(x_i - \mathbf{P}\mathbf{a}_i^\top \boldsymbol{\theta}_i)^2}{2\sigma_i^2} - \log(2\pi \sqrt{\sigma_i}) \right]. \end{aligned} \quad (3.1)$$

Taking partial derivative over $\boldsymbol{\theta}_i$, we have

$$\begin{aligned} \frac{\partial \mathcal{L}(\mathbf{x}|\Theta)}{\partial \boldsymbol{\theta}_i} &= -\frac{\mathbf{P}\mathbf{a}_i \mathbf{P}\mathbf{a}_i^\top}{\sigma_i^2} \boldsymbol{\theta}_i - \frac{x_i \mathbf{P}\mathbf{a}_i}{\sigma_i^2} \\ &\triangleq \mathbf{S}(x_i) \boldsymbol{\theta}_i + \mathbf{s}_0(x_i), \end{aligned} \quad (3.2)$$

where $\mathbf{S}(x_i)$ is a squared matrix and $\mathbf{s}_0(x_i)$ is a vector. As shown, both $\mathbf{S}(x_i)$ and $\mathbf{s}_0(x_i)$ are constant with respect to Θ . Therefore, the Fisher vector $\Phi_{\Theta}(\mathbf{x})$ is a *linear* function of Θ . This simple form of $\Phi_{\Theta}(\mathbf{x})$ significantly facilitates our further kernel learning.

3.1.2 Discriminative Fisher kernel learning via SVM—As each Fisher vector is a function of the SGBN parameters, discriminatively learning these parameters can thus be converted to learning discriminative Fisher kernels. We require that the learned SGBN models possess the following properties. Firstly, the Fisher vectors induced by the learned SGBN model should be well separated between classes. Secondly, the learned SGBN models should maintain reasonable capacity of representation. Thirdly, the learned SGBN models should not violate DAG.

We use the following strategies to achieve our goal. Firstly, to obtain a discriminative Fisher kernel, we jointly learn the parameters of SGBN and the separating hyper-plane of SVMs

²An exception [32] is discussed in ‘‘Generalization’’ in Section 3.3, which is published after our work [25].

with Fisher kernel. Radius-margin bound, the upper bound of the Leave-One-Out error, is minimized to keep good generalization of the SVMs. Secondly, to maintain reasonable representation, we explicitly control the fitting (regression) errors of the learned model during optimization. Recall that GBN learns the network by minimizing the regression errors of each node over its parent nodes. Thirdly, we enforce the DAG constraint to ensure the validity of the graph. Our method is developed as follows.

In order to use radius-margin bound, \mathcal{L}_2 -SVM with soft margin is employed [34]³, which optimizes

$$\begin{aligned} & \min_{\mathbf{w}, \boldsymbol{\xi}} \frac{1}{2} \|\mathbf{w}\|_2^2 + C \boldsymbol{\xi}^\top \boldsymbol{\xi} \\ & s.t. \ y_i(\mathbf{w}^\top \Phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \ \xi_i \geq 0, \ \forall i. \end{aligned} \quad (3.3)$$

Following the convention in SVMs, \mathbf{w} is the normal of the separating plane, b the bias term, $\boldsymbol{\xi}$ the slack variables and C the regularization parameter. Here y_i is the class label of the i -th sample. \mathcal{L}_2 -SVM can be rewritten as SVM with hard margin by slightly modifying the kernel $\mathbf{K} := \mathbf{K} + \mathbf{I}/C$, where \mathbf{I} is identity matrix. For convenience, in the following, we redefine $\mathbf{w} := [\mathbf{w}^\top \ \sqrt{C} \boldsymbol{\xi}^\top]^\top$ and $\Phi(\mathbf{x}_i) := [\Phi^\top(\mathbf{x}_i) \ \mathbf{e}_i^\top y_i / \sqrt{C}]^\top$. The vector \mathbf{e}_i has the value of 1 at the i -th element, and 0 elsewhere.

Incorporating radius information leads to solving

$$\begin{aligned} & \min_{\mathbf{w}} \frac{1}{2} R^2 \|\mathbf{w}\|_2^2 \\ & s.t. \ y_i(\mathbf{w}^\top \Phi(\mathbf{x}_i) + b) \geq 1, \ \forall i, \end{aligned} \quad (3.4)$$

where R^2 denotes the radius of Minimal Enclosing Ball (MEB). It has been observed that when the sample size is small, the estimation of R^2 may become noisy and unstable [35]. Therefore, it has been proposed to use the trace of the total scatter matrix instead for such cases [35], [36]. We finally solve the following optimization problem:

$$\begin{aligned} & \min_{\boldsymbol{\theta}, \mathbf{w}} \frac{1}{2} \text{tr}(\mathbf{S}_T) \|\mathbf{w}\|_2^2 \\ & s.t. \ y_i(\mathbf{w}^\top \Phi_{\boldsymbol{\theta}}(\mathbf{x}_i) + b) \geq 1, \ \forall i \\ & \ h(\mathbf{X}_1, \boldsymbol{\Theta}_1) \leq T_1, \ h(\mathbf{x}_2, \boldsymbol{\Theta}_2) \leq T_2, \\ & \ \boldsymbol{\Theta}_1 \in DAG, \ \boldsymbol{\Theta}_2 \in DAG. \end{aligned} \quad (3.5)$$

Here $\text{tr}(\mathbf{S}_T)$ is the trace of the total scatter matrix \mathbf{S}_T , where

$\mathbf{S}_T = \sum_{i=1}^n (\Phi(\mathbf{x}_i) - \mathbf{m})(\Phi(\mathbf{x}_i) - \mathbf{m})^\top$, and \mathbf{m} is the mean of total n samples in the kernel-induced space. It can be shown that $\text{tr}(\mathbf{S}_T) = \text{tr}(\mathbf{K}) - \mathbf{1}^\top \mathbf{K} \mathbf{1} / n$, where $\mathbf{1}$ denotes a vector

³Radius-margin bound is rooted in hard-margin SVM. \mathcal{L}_2 -SVM with soft-margin can be rewritten as SVM with hard margin.

whose elements are all 1, and \mathbf{K} the kernel matrix. Fisher vector $\Phi_{\Theta}(\mathbf{x}_j)$ is obtained as in Section 3.1.1. The function $h(\cdot)$ measures the squared fitting errors of the corresponding SGBNs for the data \mathbf{X}_1 and \mathbf{X}_2 from the two classes. It is defined as

$$h(\mathbf{X}, \Theta) = \sum_{i=1}^m \|\mathbf{x}_{:,i} - \mathbf{P}\mathbf{A}_i^{\top} \boldsymbol{\theta}_i\|_2^2. \quad (3.6)$$

The two user-defined parameters T_1 and T_2 explicitly control the degree of fitting during the learning. Adding these constraints also avoids the scaling problem of Θ .

The DAG constraint in H-SGBN could be employed to enforce the validity of the graph. However, here we adopt a new DAG constraint proposed in Section 4 due to its advantages over that of H-SGBN. The new DAG constraint employs a set of topological ordering variables (\mathbf{o}, \mathbf{T}) to guarantee DAG. It is a bilinear function of the ordering variables (\mathbf{o}, \mathbf{T}) and the SGBN parameters Θ . An elaboration is given in Section 4. At the moment, let us temporarily skip the details of this DAG constraint and concentrate on the discriminative learning.

One possible approach for solving Eqn. (3.5) is to alternately optimize the separating hyperplane \mathbf{w} and the parameter Θ . That is,

$$\begin{aligned} & \min_{\Theta, \mathbf{o}, \mathbf{T}} J(\Theta) \\ & s.t. \ h(\mathbf{X}_1, \Theta_1) \leq T_1, \ h(\mathbf{X}_2, \Theta_2) \leq T_2, \\ & \Theta_1 \in DAG(\mathbf{o}_1, \mathbf{T}_1), \ \Theta_2 \in DAG(\mathbf{o}_2, \mathbf{T}_2). \end{aligned} \quad (3.7)$$

where

$$\begin{aligned} J(\Theta) &= \min_{\mathbf{w}} \frac{1}{2} \text{tr}(\mathbf{S}_T) \|\mathbf{w}\|_2^2 \\ & s.t. \ y_i (\mathbf{w}^{\top} \Phi_{\Theta}(\mathbf{x}_i) + b) \geq 1, \quad \forall i. \end{aligned} \quad (3.8)$$

Note that for a given Θ , the term $\text{tr}(\mathbf{S}_T)$ is constant in Eqn. (3.8). Due to the strong duality in SVM optimization, we solve the term $\|\mathbf{w}\|_2^2$ by

$$\begin{aligned} J_0(\Theta) &= \max_{\boldsymbol{\alpha}} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n y_i y_j \alpha_i \alpha_j K_{\Theta}(\mathbf{x}_i, \mathbf{x}_j) \\ & s.t. \ \sum_{i=1}^n \alpha_i y_i = 0, \ \alpha_i \geq 0 \ \forall i, \end{aligned} \quad (3.9)$$

where α_j is the Lagrangian multiplier and $K_{\Theta}(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi_{\Theta}(\mathbf{x}_i), \Phi_{\Theta}(\mathbf{x}_j) \rangle$.

Algorithm 1

KL-SGBN: Discriminative Learning

Input: data $\mathbf{X}_1, \mathbf{X}_2 \in \mathbb{R}^{n \times m}$, label $\mathbf{y} \in \mathbb{R}^{n \times 1}$

Denote $\Theta = [\Theta_1, \Theta_2]$

Initialize $\Theta^{(0)}, \mathbf{o}^{(0)}, \mathbf{T}^{(0)}$ by Algorithm 3 for each class.

Let $\Theta^{(t-1)} = \Theta^{(0)}, \mathbf{o}^{(t-1)} = \mathbf{o}^{(0)}, \mathbf{T}^{(t-1)} = \mathbf{T}^{(0)}$

repeat

- 1 Compute $\Phi_{\Theta}^{(t-1)}$ and $\mathbf{K}_{\Theta}^{(t-1)}$ by Eqn. (3.2)
- 2 Compute $\text{tr}(\mathbf{S}_T)^{(t-1)} = \text{tr}(\mathbf{K}_{\Theta}^{(t-1)}) - \mathbf{1}^T \mathbf{K}_{\Theta}^{(t-1)} \mathbf{1} / n$
- 3 Solve $J_0(\Theta^{(t-1)})$ and \mathbf{a}^* by Eqn. (3.9)
- 4 $J(\Theta^{(t-1)}) = J_0(\Theta^{(t-1)}) \times \text{tr}(\mathbf{S}_T)^{(t-1)}$
- 6 Minimize Eqn. (3.7) with \mathbf{a}^* and obtain $\Theta^{(t)}$:
 - 6.1 Let $\mathbf{o} = \mathbf{o}^{(t-1)}, \mathbf{T} = \mathbf{T}^{(t-1)}$, solve $\Theta^{(t)}$ by Eqn. (3.7);
 - 6.2 Let $\Theta = \Theta^{(t)}$, solve $\mathbf{o}^{(t)}, \mathbf{T}^{(t)}$ by Eqn. (4.2).
- 7 Let $\Theta^{(t-1)} = \Theta^{(t)}, \mathbf{o}^{(t-1)} = \mathbf{o}^{(t)}, \mathbf{T}^{(t-1)} = \mathbf{T}^{(t)}$

until convergence/max number of iterations

Output: $\Theta^* = \Theta^{(t)}$

As mentioned above, the DAG constraint is a bilinear function of (\mathbf{o}, \mathbf{T}) and Θ . Many quadratic programming packages could be used to solve Eqn. (3.7). We use fmincon-SQP (sequential quadratic programming) in Matlab. Gradient information is required by many optimization algorithms (including fmincon-SQP) to speed up the line search. It is not difficult to find that the gradient of $K_{\Theta}(\mathbf{x}_i, \mathbf{x}_j)$ is just a linear function of Θ , making the evaluation of gradient $\nabla_{\Theta} J$ easy. Our learning process is summarized in Algorithm 1.

3.2 Proposed Max-margin-based Discriminative Learning (MM-SGBN)

KL-SGBN introduces group discrimination into SGBNs by optimizing the performance of SVM classifiers with SGBN-induced features. Although this leads to a relatively simple optimization problem, optimizing the performance of SVMs does not necessarily imply optimizing the discrimination of SGBNs. We believe that, the discrimination of SGBNs can be further improved if we *directly* optimize their (instead of SVMs') classification performance. Therefore we propose a new learning framework based on max-margin formulation directly built on SGBNs. We call this method MM-SGBN.

For binary classification, maximizing the minimum margin between two classes can be obtained by maximizing the minimum conditional likelihood ratio (MCLR) [18]:

$$\text{MCLR}(\Theta) = \min_{i=1}^n \frac{P(y_i | \mathbf{x}_i, \Theta_{y_i})}{P(\bar{y}_i | \mathbf{x}_i, \Theta_{\bar{y}_i})}$$

Without loss of generality, y_i and $\bar{y}_i \in \{-1, 1\}$, representing the true and false labels for the i -th sample, respectively. The parameter $\Theta_{y_i} = \Theta_1$ if $y_i = 1$, or $\Theta_{y_i} = \Theta_2$ if $y_i = -1$. We can see that MCLR identifies the most confusing sample whose probability of the true class assignment is close to or even less than that of the false class assignment. Hence, maximizing MCLR targets the maximal separation of the most confusing samples in the two classes. It is not difficult to see that MCLR can naturally handle multi-class case when replacing the denominator by the maximal probability induced by all false class assignments. Let $\Theta = [\Theta_1, \Theta_2]$. Taking log-likelihood of MCLR, we have

$$\log \text{MCLR}(\Theta) = \min_{i=1}^n (\log p(\mathbf{x}_i | y_i, \Theta_{y_i}) - \log p(\mathbf{x}_i | \bar{y}_i, \Theta_{\bar{y}_i})) + \text{const}, \quad (3.10)$$

where the prior probabilities of $P(y_i)$ and $P(\bar{y}_i)$ that are irrelevant to Θ are absorbed into the constant term. Eqn. (3.10) can be shown to be a quadratic function of Θ in the case of SGBN. In order to maximize MCLR, we require the difference of log-likelihood function in Eqn. (3.10) be larger than a margin for all samples, i , and maximize the margin r . To deal with hard separations, we employ a soft margin formulation as follows.

$$\min_{\Theta_1, \Theta_2, \xi_i, r, \mathbf{o}, \Upsilon} \lambda \sum_{i=1}^n \xi_i - r \quad (3.11)$$

$$\text{s.t. } y_i (\mathcal{L}(\Theta_1, \mathbf{x}_i) - \mathcal{L}(\Theta_2, \mathbf{x}_i)) \geq r - \xi_i, \quad \forall i \quad (3.11a)$$

$$\xi_i \geq 0, \quad r \geq 0, \quad (3.11b)$$

$$h(\mathbf{X}_1, \Theta_1) \leq T_1, \quad h(\mathbf{X}_2, \Theta_2) \leq T_2 \quad (3.11c)$$

$$\Theta_1 \in \text{DAG}(\mathbf{o}_1, \Upsilon_1), \quad \Theta_2 \in \text{DAG}(\mathbf{o}_2, \Upsilon_2) \quad (3.11d)$$

The constraints in (3.11a) enforce the likelihood of \mathbf{x}_i to its true class larger than that to its false class by a margin r . The variables ξ_i are slack variables indicating the intrusion of the margin. The function $\mathcal{L}(\cdot)$ denotes the log-likelihood, defined in Eqn. (3.1). We require $\mathcal{L}(\Theta_1, \mathbf{x}_i)$ larger than $\mathcal{L}(\Theta_2, \mathbf{x}_i)$ when $y_i = 1$, and $\mathcal{L}(\Theta_2, \mathbf{x}_i)$ larger than $\mathcal{L}(\Theta_1, \mathbf{x}_i)$ when $y_i = -1$.

Algorithm 2

MM-SGBN: Discriminative Learning

Input: data $\mathbf{X}_1, \mathbf{X}_2 \in \mathbb{R}^{n \times m}$, label $\mathbf{y} \in \mathbb{R}^{n \times 1}$

Denote $\Theta = [\Theta_1, \Theta_2]$

Initialize $\Theta^{(0)}, \mathbf{o}^{(0)}, \mathbf{T}^{(0)}$ by Algorithm 3 for each class.

Fix $\Theta = \Theta^{(0)}$ and estimate $r^{(0)}$ and $\xi_i^{(0)}$ by Eqn. (3.11)

only with the two constraints (3.11a) and (3.11b).

Initialize $t = 1$.

repeat

Step 1: Fixing $\mathbf{o} = \mathbf{o}^{(t-1)}$ and $\mathbf{T} = \mathbf{T}^{(t-1)}$, optimize Eqn. (3.11) with the constraints (3.11a ~ 3.11c) to update $\Theta^{(t)}$, $r^{(t)}$ and $\xi_i^{(t)}$;

Step 2: Fixing $\Theta^{(t)}$, optimize Eqn. (4.2) to update $\mathbf{o}^{(t)}$ and $\mathbf{T}^{(t)}$ to enforce DAG.

Let $t = t + 1$

until convergence/max number of iterations

Output: $\Theta^* = \Theta^{(t)}$

The constraints in (3.11c) control the fitting errors, same to that used in KL-SGBN, and the function $h(\cdot)$ is defined in Eqn. (3.6).

The constraints in (3.11d) are the DAG constraint proposed in Section 4, Eqn. (4.1). To enforce the validity of DAG on both graphs, we introduce a set of order variables $\mathbf{o} = \{o_1, o_2, \dots, o_m\}$ and \mathbf{T} for each class separately, and employ the constraints stated in Eqn. (4.1). Please refer to Eqn. (4.1) for details.

The optimization in Eqn. (3.11) can be solved iteratively by optimizing (Θ, ξ_i, r) and (\mathbf{o}, \mathbf{T}) alternately, as summarized in Algorithm 2. In Step 1, we solve a linear objective function with n non-convex and two convex quadratic constraints by fmincon-SQP (sequential quadratic programming) in Matlab. In Step 2, we solve the linear programming by the package of CVX⁴.

It is worthy noting that, we learn an SGBN model for each individual class in order to meet the requirement of both interpretation and discrimination in exploratory research. For example, each SGBN may model the brain network of the healthy or the diseased class, as well as carrying the essential class discrimination. Both the network modelling and the discrimination are of interest in such cases. Our method is different from the conventional BN classifiers [16], [17], [18] that solely focus on classification. In those methods, only a single BN is learned to reflect the “difference” of the two classes. It does not model any individual class as our method does, and hence deviates from our purpose of both representing and discriminating brain networks. Moreover, the works in [16], [17], [18]

⁴<http://cvxr.com/cvx/>

cannot handle the continuous variables of brain imaging measures, and inherit the drawbacks of the traditional two-stage methods.

3.3 Discussion and Analysis

In the following, some issues regarding the two proposed discriminative learning frameworks are discussed.

Classifiers: The proposed discriminative learning frameworks produce a set of jointly learned SGBN models, one for each class. Based on these SGBN models, two kinds of classifiers can be constructed, i.e., the SGBN classifier and the SVM classifier. The SGBN classifier categorizes a sample by comparing the sample's likelihood according to each SGBN model. The SVM classifier is trained by the sample-specific Fisher vectors induced from the SGBN models. These two classifiers are tightly coupled by the underlying SGBN models. Specifically, more discriminative SGBN models directly lead to a better SGBN classifier, and can provide discriminative Fisher vectors to SVM for better classification. Rooted in this relationship, both the KL-SGBN and the MMSGBN can improve the classification performance of these two classifiers simultaneously. Put simply, KLSGBN explicitly optimizes the SVM classifier and in turn implicitly improves the SGBN classifier; while MMSGBN explicitly optimizes the SGBN classifier, bringing an implicit improvement of the SVM classifier as well. When evaluating the discriminative power of the learned SGBN models by the SGBN classifier (a direct measurement), it is therefore expected that MM-SGBN can outperform KL-SGBN. However, KL-SGBN has some computational advantages and provides a new perspective to manipulate BN models, analyzed as follows.

Computational Issues: Compared with KL-SGBN, MM-SGBN requires to solve more complicated optimization problems, which may become problematic when the number of training samples increase. Let us compare Eqn. (3.7) for KL-SGBN and Eqn. (3.11) for MMSGBN. For KL-SGBN, Eqn. (3.7) optimizes $\mathcal{J}(\Theta)$ with two convex quadratic constraints of data fitting and two DAG constraints, which are independent of the number of training samples n . The evaluation of $\mathcal{J}(\Theta)$ needs to solve an SVM-like problem in Eqn. (3.8), taking just n linear constraints of Θ , which could be efficiently solved by off-the-shelf SVM packages. For MM-SGBN, in addition to the data fitting and DAG constraints as in Eqn. (3.7), the optimization problem in Eqn. (3.11) also has to satisfy n non-convex quadratic constraints. When n increases to a medium or large value, the optimization problem could be quite hard to solve.

Edge Selection: In addition to the discriminative learning of SGBN, the employment of Fisher kernel in KL-SGBN also provides a new perspective of edge selection for GBN. As introduced in Section 3.1.1, applying Fisher kernel on GBN produces sample-specific feature vectors whose component is the gradient of the log likelihood, i.e., $\frac{\partial \mathcal{L}(\mathbf{x}|\Theta)}{\partial \Theta_{ij}}$. In other words, each feature now corresponds to an edge Θ_{ij} in the SGBN. This makes it possible to convert the SGBN edge selection to a more traditional feature selection problem that has been well studied and has a large body of options in the literature. Edge selection has been

employed in our work to deal with the “small sample size” problem that is often encountered in medical applications. For example, it is common to have only 100 training samples but 3200 parameters (for SGBNs of 40 nodes from two classes) to learn in brain network analysis. To handle this issue, we keep using the whole Θ for computing \mathbf{K}_Θ , but only optimize a selected subset Θ_s . There are many options to determine Θ_s . We just compute the Fisher vector Φ_Θ for each sample, calculate the Pearson correlation between each component of Φ_Θ and the class labels on the training data, and select the top θ_j with the highest correlations. To keep our problem simple, only the parameters associated with edges present in the graph are optimized to avoid the violation of DAG. It is remarkable that even this simple selection process has significantly improved the discrimination for both KL-SGBN and MM-SGBN. Note that this edge selection step is essentially different from that of the traditional two-stage methods. It is just an empirical method to handle the small sample size problem and will become *unnecessary* when sufficient training data are available. In contrast, identifying the candidate-parent sets is an indispensable step in two-stage methods to obtain computationally tractable solutions.

Generalization: We would like to point out that our learning framework of KL-SGBN could be easily generalized. It could be used to discriminatively learn the parameters of distributions other than that represented by GBN by just simply switching GBN to the target distribution, such as Gaussian Mixture Model (GMM). Indeed, this has been seen in [32], after our work [25]. However, as shown in this paper, the Fisher vector of GBN is a linear function of the model parameters, which significantly simplifies the learning problem. This favorable property may not be guaranteed with other distributions, including GMM.

4 Proposed DAG Constraint

In this section, we revisit H-SGBN and propose a new DAG constraint that could simplify the optimization problems in SGBN and its discriminative learning process as introduced in Sections 3.1 and 3.2.

4.1 H-SGBN Revisited

Recall that, the DAG constraint in H-SGBN (Section 2.1) utilizes the matrix \mathbf{P} , an implicit function of Θ , which significantly complicates the optimization problem in Eqn. (2.1). In [14], for simplicity, in each optimization iteration, \mathbf{P} is first treated as a constant while optimizing Θ , and then recalculated by searching on the binarized new Θ . This hard binarization could introduce high discontinuity of Θ into the optimization. Solving Θ column-wisely by BCD may mitigate this problem since only one column of Θ is changed in each iteration, inducing less discontinuity. However, we observe that the solution of BCD depends on which column of Θ to be optimized first. In other words, if we randomly permute the ordering of features (the columns in \mathbf{X}), we will obtain different SGBNs, which impairs the interpretability of the SGBN model. The optimization ordering matters because the matrix \mathbf{P} used in the DAG constraint changes with the ordering. This problem has been demonstrated in our experiment. Moreover, we find experimentally that if \mathbf{P} is solved as a whole instead of BCD, the optimization in Eqn. (2.1) will not converge but oscillate between some *non-DAG* solutions, possibly due to the high discontinuity mentioned above⁵. Early

stop cannot help because no premature solution satisfies DAG. These optimization difficulties motivate our work of proposing a new DAG constraint that is much simpler for SGBN, as described below.

4.2 Proposed DAG constraint

It is known that, a BN is equivalent to a topological ordering (Page 362 in [37]). Therefore, we propose a new DAG constraint applicable to continuous variables with GBN based on this equivalence. With a few linear inequalities and variables separable from Θ , the new DAG constraint significantly simplifies that used in [14]. Specifically, given a directed graph \mathcal{G} and the parameters Θ , a real-valued order variable o_i is assigned to each node i , where $0 < o_i < \Delta$, and Δ is a predefined arbitrary positive number. We propose a sufficient and necessary condition for \mathcal{G} to be DAG as in Proposition 1.

Proposition 1: Given a sparse Gaussian Bayesian Network parameterized by Θ and its associated directed graph \mathcal{G} with m nodes, the graph \mathcal{G} is DAG if and only if there exist some $o_i (i = 1, \dots, m)$ and $\Upsilon \in \mathbb{R}^{m \times m}$, such that for arbitrary $\Delta > 0$, the following constraints are satisfied:

$$o_j - o_i \geq \frac{\Delta}{m} - \Upsilon_{ij}, \quad \forall i, j \in \{1, \dots, m\}, \quad i \neq j \quad (4.1a)$$

$$\Upsilon_{ij} \geq 0, \quad (4.1b)$$

$$\Upsilon_{ij} \times \Theta_{ij} = 0, \quad (4.1c)$$

$$\Delta \geq o_i \geq 0. \quad (4.1d)$$

Eqn.(4.1) leads to a topological ordering equivalent to DAG. The topological ordering means that if node j comes after node i in the ordering ($o_j > o_i$), there cannot be a link from node j to node i , which guarantees the acyclicity. The proof of Proposition 1 is given in Appendix.

By Proposition 1, we remove the awkward hard binarization for computing \mathbf{P} in [14]. The inequalities of (4.1a, 4.1b, 4.1d) are linear to the ordering variables o_i and Υ . The equation (4.1c) differs from the equation $\Theta_{ji} \times \mathbf{P}_{jj} = 0$ in [14] in that the variable Υ_{ij} is now separable from Θ_{ij} (while \mathbf{P}_{jj} is not) and does not require the binarization of Θ . This makes it tractable to solve Θ as a whole instead of BCD (to avoid the feature ordering problem).

⁵Please note that, solving Θ column-wisely without updating \mathbf{P} in each iteration will only lead to non-DAG solutions

It is worth noting that, provided Θ is sparse, the number of constraints in Eqn. (4.1) could be significantly reduced. As can be seen, for any $\Theta_{ij} = 0$, as long as we set the corresponding Υ_{ij} an arbitrary value greater than $(\frac{1}{m}+1)\Delta$, all the conditions in Eqn. (4.1) will be automatically satisfied. Therefore, we only need to consider the constraints related to $\Theta_{ij} \neq 0$.

The idea of topological ordering is also used to design DAG constraint for the discrete variables in [38]. However, the work in [38] addresses the multinomial distribution of discrete variables, while here we target the Gaussian distribution of continuous variables. It is worthy noting that the constraint in [38] has to predefine candidate parent-node sets. Therefore, it inherits the drawbacks of the two-stage methods as pointed out in Section 1. This has been circumvented in our proposed DAG constraint for SGBN.

4.3 Estimation of SGBN from A Single Class

With our DAG constraint proposed in Eqn. (4.1), we could estimate SGBN from a single class as the initial solution to our discriminative learning of KL-SGBN or MM-SGBN. In particular, we optimize

$$\begin{aligned} \min_{\Theta, \mathbf{o}, \Upsilon} \sum_{i=1}^m \|\mathbf{x}_{:,i} - \mathbf{P}\mathbf{A}_i^\top \boldsymbol{\theta}_i\|_2^2 + \lambda_1 \|\boldsymbol{\theta}_i\|_1 + \lambda_{dag} \boldsymbol{\varepsilon}_i^\top |\boldsymbol{\theta}_i| \\ \text{s.t. } o_j - o_i \geq \frac{\Delta}{m} - \Upsilon_{ij}, \forall i, j \in \{1, \dots, m\}, i \neq j \quad 0 \leq o_i \leq \Delta, \Upsilon_{ij} \geq 0, \end{aligned} \quad (4.2)$$

where $\boldsymbol{\varepsilon}_i$ is the i -th column of the matrix \mathbf{T} , and $|\boldsymbol{\theta}_i|$ the component-wise absolute value of $\boldsymbol{\theta}_i$. This optimization problem is solved in an iterative way with two alternate steps in each iteration: i) optimize \mathbf{o} and Υ (with Θ fixed) and ii) optimize Θ (with \mathbf{o} and Υ fixed). This process is repeated until convergence. We call this proposed method OR-SGBN (Algorithm 3).

When the coefficient λ_{dag} is sufficiently large, the alternate optimization strategy of Eqn. (4.2) will converge to a DAG solution, as shown in Proposition 2 in Appendix. In practice, for numerical stability, we adopt a ‘‘warm start’’ strategy as in [14], that is, to gradually increase the values of λ_{dag} until the resulting \mathcal{G} becomes DAG. Specifically, we use a set of values of $\lambda_{dag}: \lambda_{dag}^{(1)} < \lambda_{dag}^{(2)} < \dots < \lambda_{dag}^{(M)}$ to solve Eqn. (4.2) (Algorithm 3).

We use a bias variable $x_0 = 1$ in the regression model to improve data fitting, thus

$x_i = [\boldsymbol{\theta}_i \quad \theta_0]^\top [\mathbf{P}\mathbf{a}_i \quad 1] + \varepsilon_i (i > 1)$. In the following part, we denote $\boldsymbol{\theta}_i \triangleq [\boldsymbol{\theta}_i \quad \theta_0]$ and $\mathbf{P}\mathbf{a}_i \triangleq [\mathbf{P}\mathbf{a}_i \quad 1]$. The bias term θ_0 is learned together with other $\boldsymbol{\theta}_i$. This equals to introducing a bias node into the graph. It has no parent but is the parent of all the other nodes. If the original graph is a DAG, this does not cause the violation of DAG.

It is interesting yet challenging to analyze the network consistency of OR-SGBN. It is noted that Eqn. (4.2) can be reorganized into a weighted LASSO problem, which can be conceptually linked to ‘‘adaptive LASSO’’ in the literature [39], [40], [41]. The analysis framework provided by these works is suggestive of promising strategies to analyze the network consistency for L1-penalized Gaussian networks. However, a complete treatment of

this analysis for OR-SGBN requires a deep investigation. Considering the significant amount of the required workload and its importance, we will explore this problem in a separate paper in our future work.

Algorithm 3

OR-SGBN: SGBN from a single class

Input: data $\mathbf{X} \in \mathbb{R}^{n \times m}$

Initialize $\Theta^{(0)}$ by least square fitting.

Initialize $\mathbf{o}^{(0)}$ and $\mathbf{T}^{(0)}$ by solving Eqn. (4.2) with $\Theta = \Theta^{(0)}$.

Let $T = 1$.

repeat

Fixing $\mathbf{T} = \mathbf{T}^{(T-1)}$ and $\mathbf{o} = \mathbf{o}^{(T-1)}$.

Let $t = 1$, $\Theta^{(T-1,t=0)} = \Theta^{(T-1)}$.

for $\lambda_{dag} = \lambda_{dag}^{(1)}$ **to** $\lambda_{dag}^{(M)}$ **do**

Optimize Eqn. (4.2) with the initial solution $\Theta^{(T-1,t=1)}$ to obtain $\Theta^{(T-1,t)}$.

Let $t=t+1$.

end for

Let $\Theta^{(T)} = \Theta^{(T-1,M)}$.

Fixing $\Theta^{(T)}$, optimize Eqn. (4.2) to update $\mathbf{o}^{(T)}$ and $\mathbf{T}^{(T)}$ to enforce DAG.

Let $T = T + 1$.

until convergence/max number of iterations

Output: $\Theta^* = \Theta^{(T)}$

5 Experiment

In this section, we investigate the properties of our proposed methods from three aspects: the DAG constraint, the discriminative learning process, and the resulting connectivity for brain network analysis. Four experiments are conducted, summarized in Table 2. The data sets and the experiments are elaborated as follows.

5.1 Neuroimaging Data Sets

We conduct our experiment on the publicly accessible ADNI [42] database to analyze brain effective connectivity for the Alzheimer's disease. Three data sets are used from two imaging modalities of MRI and FDG-PET downloaded from ADNI. They are elaborated as follows.

MRI data set includes 120 T1-weighted MR images belonging to 50 mild cognitive impairment (MCI) patients and 70 normal controls (NC). These images are preprocessed by the typical procedure of intensity correction, skull stripping, and cerebellum removal. We segment the images into gray matter (GM), white matter (WM), and cerebrospinal fluid (CSF) using the standard FSL⁶ package, and parcellate them into 93 Region of Interest

⁶<http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/>

(ROI) based on an ROI atlas [43] after spatial normalization. The GM volume of each ROI is used as the imaging feature to characterize each network node. Forty ROIs are included in this study, following [14]. They have higher correlation with the disease and are mainly located in the temporal lobe and subcortical region. Studying brain morphology as a network can take the advantage of statistical tools from graph theory. Moreover, it has been reported that the covariation of gray matter morphology might be related to the anatomical connectivity [44].

PET data set includes 103 FDG-PET images (and their corresponding MR images) of 51 AD patients and 52 NC. The MR images belonging to different subjects are co-registered and partitioned into ROIs as before. The ROI partitions are copied onto their corresponding PET images by a rigid transformation. The average tracer uptakes within each ROI is used as the imaging feature to characterize each network node. Forty ROIs discriminative to the disease are used in the study. The retention of tracer in FDG-PET is analogous to the glucose uptake, thus reflecting the tissue metabolic activity.

MRI-II data set is similar to the MRI data set but using 40 different ROIs covering the typical brain regions spread over the frontal, parietal, occipital and temporal lobes.

We randomly partition each data set into 30 groups of training-test pairs. Each group includes 80 training and 40 test samples in MRI and MRI-II, or 60 training and 43 test samples in PET.

5.2 DAG Constraint

With our proposed DAG constraint, the SGBN model for an individual class can be learned with all the parameters Θ optimized together (OR-SGBN), instead of column-wisely as did in [14], [25], [26]. To explore the properties of our DAG constraint, we test three experimental configurations, namely, OR-SGBN (WHOLE), H-SGBN (BCD) and H-SGBN (WHOLE). The word in the parenthesis is used to explicitly indicate whether the parameters Θ are optimized together (WHOLE) or column-wisely (BCD). OR-SGBN (WHOLE) is our SGBN learning method for a single class in Algorithm 3, implemented with the package of CVX. H-SGBN (BCD) is the column-wise method in [14] and implemented with the code downloaded from the authors' website. H-SGBN (WHOLE) is our attempt to optimize Θ together for the objective function of H-SGBN in [14], which is implemented with the package of CVX⁷. The same Θ that is computed by a sparse least square fitting of the training set is provided to all the methods to initialize the optimizations. The "warm-start" strategy is applied wherever applicable in all methods.

It is found that when solving all Θ as a whole, H-SGBN (WHOLE) that uses the DAG constraint in [14] does not converge: the optimization is trapped to oscillate between a few solutions that are not DAG. Therefore, from now on, we only consider H-SGBN (BCD) and OR-SGBN (WHOLE).

⁷The optimization problem is solved by a series of convex subproblems.

Exp-I: In this experiment, we compare the solutions of OR-SGBN (WHOLE) and H-SGBN (BCD) with respect to the change of feature ordering. To do that, for the neuroimaging data sets, we randomly permute the feature ordering for 100 times. The estimated Θ of the resulting 100 SGBNs are re-arranged according to the initial feature ordering and then averaged as in Fig. 2. As shown, the averaged result from OR-SGBN (WHOLE) (Fig. 2(d)) is almost identical to the result using the original feature ordering (Fig. 2(c)), reflecting its robustness to feature ordering. In contrast, H-SGBN (BCD) generates SGBNs with large variations when the feature ordering changes ((Fig. 2(a) versus (b)). To give a quantitative evaluation, the Euclidean distance and the correlation between the averaged Θ and the original Θ are presented in Table 3. Consistently, the solutions from OR-SGBN (WHOLE) are much less affected by the ordering permutation, indicating the advantage of solving Θ as a whole via the proposed DAG constraint.

Exp-II: In this experiment, we test the ability of OR-SGBN (WHOLE) at identifying network structures from data. Since no ground-truth is available for the three neuroimaging data sets due to the unknown mechanism of the disease, we conduct experiments on nine benchmark network data sets mostly coming from the Bayesian Network Repository [45] as was done in the literature [12], [46]. The nine benchmark data sets are: Factors (27 nodes, 68 arcs), Alarm (37 nodes, 46 arcs), Barley (48 nodes, 84 arcs), Carpo (61 nodes, 74 arcs), Chain (7 nodes, 6 arcs), Hailfinder (56 nodes, 66 arcs), Insurance (27 nodes, 52 arcs), Mildew (35 nodes, 46 arcs) and Water (32 nodes, 66 arcs). We compare the OR-SGBN (WHOLE) with another eight BN learning methods, including LIMB [12], GS [47], TC and its variant TC-bw [13] and three variants of IAMB [48]. The experiment is repeated for 50 simulations. In each simulation, for each network, we randomly sample 1000 samples from $\pm\text{Uniform}(0.5, 1)$ for the regression coefficients of each variable on its parents. The parameters of the eight methods to be compared are set according to [14]. A predefined λ that controls the sparsity of OR-SGBN is uniformly applied to all the nine data sets, which simply brings the number of the resulting edges to a reasonable range⁸. We use the first stage estimate of LIMB as the initial solution of OR-SGBN. Table 4 shows the total numbers of mis-identified edges (including both the false and the missing edges), while Table 5 shows the numbers of falsely identified edges (false positive). In addition, Table 6 lists the numbers of falsely identified PDAG structures. PDAG structures are statistically indistinguishable structures, i.e., representing the same statistical dependency. The PDAG of BN is obtained by the method in [49]. From Tables 4 ~ 6, it can be seen that OR-SGBN shows significantly smaller errors on six data sets (Factors, Alarm, Barley, Carpo, Hailfinder and Insurance) in identifying both edges and PDAG structures. For the data sets of Mildew and Water, OR-SGBN performs similarly to the other methods. It only performs relatively inferior on Chain. This experiment demonstrates that the proposed DAG constraint for SGBN can perform effectively for BN structure identification. Its relatively low risk of mis-edge identification is a favorable property for exploratory research.

⁸The Bayesian Information Criterion is used to select λ in [14]. However, it did not behave well in our experiment.

5.3 Comparison of Discrimination

After testing the effectiveness of the proposed DAG constraint, we now investigate the theme of this paper: the discriminative learning frameworks. We consider two kinds of classifiers: i) the SGBN classifier (with two SGBN models, one for each class), and ii) the SVM classifier learned by the Fisher vectors induced from the SGBN models.

Exp-III: In this experiment, we test whether our learning methods (KL-SGBN and MM-SGBN) can improve the discriminative power on both kinds of classifiers for the real neuroimaging data sets. The initial SGBN models are obtained by our proposed OR-SGBN (WHOLE), since it has been shown more robust to feature ordering than H-SGBN as above. For the SGBN classifier, assuming equal prior, we assign a test sample to the class with a higher likelihood. For the SVM classifier, we use \mathcal{L}_2 -SVM with Fisher kernels. In order to maintain representation capability, we allow maximal 1% additional squared fitting errors (that is, $T_i = 1.01 \times T_{\lambda}$, ($i = 1, 2$), where T_{λ} is the squared fitting error of the initial solution) to be introduced during the learning process of KL-SGBN or MM-SGBN.

The test accuracies are averaged over the 30 randomly partitioned training-test groups. The classification performances of SGBN and SVM classifiers are evaluated with the varied parameter λ that controls the sparsity level and the number of edges optimized in the learning process in Fig. 3. The results of our proposed KL-SGBN and MM-SGBN are plotted by the green and the red lines, respectively. The results of the individually learned OR-SGBN and H-SGBN are plotted by the blue and the black lines, respectively. The top two rows in Fig. 3 correspond to the results from the SGBN classifiers, while the bottom two rows correspond to those from the SVM classifiers. From Fig. 3, we have the following observations.

- i. Both KL-SGBN and MM-SGBN can significantly improve the discriminative power of the individually learned SGBNs (Fig. 3, the top two rows), as well as their associated SVM classifiers (Fig. 3, the third row). Such improvements are consistent over the three neuroimaging data sets and different parameter settings, and could reach the significant increases of 10% ~ 20% on most occasions. When the network becomes more sparse, the classification performance of the individually learned SGBNs (H-SGBN and OR-SGBN) drops significantly possibly due to the insufficient modeling of data. However, under such circumstances, KL-SGBN and MM-SGBN can still maintain high classification accuracies, which may indicate the necessity and effectiveness of the discriminative learning in classification scenarios.
- ii. When using SGBN classifiers, for all the three data sets, MM-SGBN consistently achieves higher test accuracies at all sparsity levels (Fig. 3, the first row) with different numbers of optimized edges than KL-SGBN (Fig. 3, the second row). The advantage of MM-SGBN over KL-SGBN comes from explicitly optimizing the discriminative power of SGBNs, instead of getting help from optimizing the performance of SVM on SGBN-induced features.

- iii. When using SVM classifiers, the SVM built upon KL-SGBN-induced features performs slightly better than that built upon MM-SGBN-induced features at all sparsity levels (Fig. 3, the third row). This is expected since KL-SGBN optimizes the performance of its associated SVM classifier.
- iv. When cross-referencing the first and the third rows in Fig. 3, it is noticed that SVM classifiers in general perform worse than the discriminatively learned SGBN classifiers. These may be because our Fisher vectors have very high dimensionality, which causes serious overfitting of data in SVM classifiers. Such situation might be somewhat improved for SGBN-classifiers since the simple Gaussian model may “regularize” the model fitting. Based on this assumption, we further select a number of leading features from Fisher vectors by computing the Pearson correlation of the features and the labels, and use the selected features to construct the Fisher kernel for the SVM classifiers. As shown in the fourth row of Fig. 3, the simple feature selection step can further significantly improve the classification performance of the Fisher-kernel based SVM.
- v. The individually learned OR-SGBN and H-SGBN perform similarly for classification. However, as mentioned above, OR-SGBN has an additional advantage of being invariant to the feature ordering.
- vi. Recall that these improvement on discrimination are achieved with no more than 1% increase of squared fitting errors, which is explicitly controlled through the user defined parameters T_1 and T_2 . Note that the rate of 1% is application dependent. More tolerance of fitting errors can potentially bring more discrimination.

5.4 Comparison of Connectivity

We also conduct an investigation to gain some insights into the learned brain networks for the diseased and the healthy populations, respectively.

Exp-IV: In this experiment, we visualize the learned brain networks and compare the connectivity patterns obtained by different methods and from different populations. MRI-II data set is used for this study since it covers regions spread over the four lobes of brain.

The structures of the brain networks recovered from NC and MCI groups are displayed in Fig. 4 by using H-SGBN (BCD) and OR-SGBN (WHOLE), respectively. The network structure is obtained by thresholding the edge weights Θ with a cutoff value of 0.01 [14]. Each row i represents the effective connections (dark dots) starting from the node i , and each column j represents the effective connections ending at the node j . Note that, due to the different optimization problems involved, the same parameter λ leads to different sparsity levels for H-SGBN and OR-SGBN. However, for a given method, different λ values do not change the major structures of the resulting networks.

In Fig. 4, it is noticed that H-SGBN (BCD) usually generates more connections in the upper triangle of the graphs even when we randomly permute the nodes. We suspect that this is caused by the column-wise optimization. The parameters θ_j (corresponding to the columns

in the graph) optimized at the early stage tend to be made more sparse than those optimized later in order to satisfy the DAG constraint. This phenomenon is not observed in OR-SGBN (WHOLE) that is used to initialize the discriminative learning.

Let us focus on OR-SGBN. Compared with H-SGBN, OR-SGBN has an additional bias node corresponding to the last row and column in Fig. 4. Visualizing Θ can provide rich information for medical analysis. Here we just list a few observations as examples. With the same λ , OR-SGBN produces 183 edges for NC, and 145 edges for MCI. Such loss of connectivity also happens at the temporal lobe (24% loss) for MCI. The temporal lobe (and some subcortical structures) is known to play a very important role in the progression of AD. The loss of connectivity in this region has been well-documented in wide AD-related studies [50], [51], [14]. In Fig. 4, we also observe an increase of connectivity (the left bottom corner in the figure) between the frontal and the temporal lobe in MCI. Some study [52] mentioned that the frontal lobe may have connectivity increase at the early stage of AD as a compensation of cognitive functions for the patients. Moreover, significant directionality changes are also found for the left (node 35) and the right (node 38) hippocampi, an important structure among the earliest ones affected by AD. Both hippocampi have reduced incoming connectivity (communications dominated by other nodes) but increased outgoing connectivity (communications dominated by themselves) in MCI. Please note that the above observations could be influenced by the factors such as the limited number of data, the degree of disease progression and the imaging modality used in this study. More reliable medical analysis should be validated on larger data sets and worth further exploration, which is, however, beyond the scope of this paper.

To illustrate the difference of edge weights learned by KL-SGBN and MM-SGBN, an example of 30 edge weight changes (from the initial OR-SGBN) learned by these two methods is given in Fig. 5, where the SGBN networks from the two classes are vectorized and concatenated as x -axis. As shown, the signs of weight changes are quite similar in both methods. The most significant difference is that, MM-SGBN gives negative weight changes to the bias node of the left Amygdala and the right Parahippocampus (red lines in Fig. 5) while KL-SGBN gives them positive weight changes. The adjustment of edge weights leads to 10% increase of test accuracy for MM-SGBN in this example.

6 Conclusion

In this paper, we focus on the discriminative learning of Bayesian network for continuous variables, especially for neuroimaging data. Two discriminative learning frameworks are proposed to achieve this goal, i.e., KL-SGBN improves the performance of SVM classifiers based on SGBN-induced features, and MM-SGBN explicitly optimizes an SGBN-based criterion for classification. We demonstrate how to utilize Fisher-kernel to bridge the generative methods of SGBN and the discriminative classifiers of SVM, and how to embed the max-margin criterion into SGBN for discriminative learning. The optimization problems are analyzed in details, and the advantages and disadvantages of the proposed methods are discussed. Moreover, a new DAG constraint is proposed to ensure the validity of the graph with theoretical guarantee and validation on the benchmark data. We apply the proposed methods to modeling brain effective connectivity for early AD prediction. Significant

improvements have been observed in the discriminative power of both the SGBN models and the associated SVM classifiers, without sacrificing much representation power.

References

1. Verma T, Pearl J. Equivalence and synthesis of causal models. *Uncertainty in Artificial Intelligence*. 1991; 6:255–268.
2. Spirtes P, Glymour C. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*. 1991; 9(1):62–72.
3. Fast, A. PhD thesis. University of Massachusetts Amherst; 2010. Learning the structure of Bayesian networks with constraint satisfaction.
4. Scutari, M. Bayesian network constraint-based structure learning algorithms: Parallel and optimised implementations in the bnlearn r package. *CoRR*; 2014.
5. Friedman N, Koller D. Being bayesian about network structure - bayesian approach to structure discovery in bayesian networks. *Machine Learning*. 2003; 50(1-2):95–125.
6. Koivisto M, Sood K. Exact bayesian structure discovery in bayesian networks. *Journal of Machine Learning Research*. 2004; 5:549–573.
7. Geiger, D., Heckerman, D. Learning gaussian networks. *CoRR*; 2013.
8. Suzuki, J. A construction of bayesian networks from databases based on an mdl principle. *UAI*; 1993. p. 266-273.
9. Acid S, Campos L. Searching for bayesian network structures in the space of restricted acyclic partially directed graphs. *Journal of Artificial Intelligence Research*. 2003; 18:445–490.
10. Tsamardinos I, Brown L, Aliferis C. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*. 2006; 65(1):31–78.
11. Gamez, Jose A., JLM, Puerta, JM. Learning bayesian networks by hill climbing: Efficient methods based on progressive restriction of the neighborhood. *Data Mining and Knowledge Discovery*. 2011; 22(1-2):106–148.
12. Schmidt, M., Niculescu-Mizil, A., Murphy, K. Learning graphical model structures using l1-regularization paths. *AAAI*; 2007.
13. Pellet, J., Elisseeff, A. Using markov blankets for causal structure learning. *JMLR*; 2008. p. 1295-1342.
14. Huang S, Li J, Ye J, Fleisher A, Chen K, Wu T, Reiman E. A sparse structure learning algorithm for gaussian bayesian network identification from high-dimensional data. *IEEE TPAMI*. 2013; 35(6):1328–1342.
15. Xiang, J., Kim, S. A* lasso for learning a sparse bayesian network structure for continuous variables. *NIPS*; 2013. p. 2418-2426.
16. Pernkopf F, Bilmes J. Efficient heuristics for discriminative structure learning of bayesian network classifiers. *JMLR*. 2010; 11:2323–2360.
17. Pernkopf F, Wohlmayr M, Tschitschek S. Maximum margin bayesian network classifiers. *IEEE TPAMI*. 2012; 34(3):521–532.
18. Guo, Y., Wilkinson, D., Schuurmans, D. Maximum margin bayesian networks. *UAI*; 2005.
19. Jaakkola, T., Haussler, D. Exploiting generative models in discriminative classifiers. *NIPS*; 1998.
20. Bullmore E, Sporns O. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nat Rev Neurosci*. 2009; 10(3):186–198. [PubMed: 19190637]
21. Smith S, Miller K, Khorshidi G, Webster M, Beckmann C, Nichols T, Ramsey J, Woolrich M. Network modelling methods for fmri. *Neuroimage*. 2011; 54(2):875–891. [PubMed: 20817103]
22. Li R, Yu J, Zhang S, Bao F, Wang P, Huang X, Li J. Bayesian network analysis reveals alterations to default mode network connectivity in individuals at risk for alzheimer’s disease. *PLoS One*. 2013; 8(12):e82104. [PubMed: 24324753]
23. Li R, Wu X, Chen K, Fleisher A, Reiman E, Yao L. Alterations of directional connectivity among resting-state networks in alzheimer disease. *Am J Neuroradiol*. 2012

24. Li X, Coyle D, Maguire L, Watson D, McGinnity T. Gray matter concentration and effective connectivity changes in alzheimers disease: A longitudinal structural mri study. *Neuro-radiology*. 2011; 53(10):733–748.
25. Zhou, L., Wang, L., Liu, L., Ogunbona, P., Shen, D. Discriminative brain effective connectivity analysis for alzheimers disease: A kernel learning approach upon sparse gaussian bayesian network. *CVPR*; 2013.
26. Zhou, L. Max-margin based learning for discriminative bayesian network from neuroimaging data. *MICCAI*; 2014.
27. Kim J, Zhu W, Chang L, Bentler P, Ernst T. Unified structural equation modeling approach for the analysis of multisubject, multivariate functional mri data. *Human Brain Mapping*. 2007; 28:85–93. [PubMed: 16718669]
28. Friston K, Harrison L, Penny W. Dynamic causal modeling. *Neuroimage*. 2003; 19:1273–1302. [PubMed: 12948688]
29. Perina A, Cristani M, Castellani U, Murino V, Jovic N. Free energy score spaces: Using generative information in discriminative classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2012; 34(7):1249–1262. [PubMed: 22156097]
30. Perronnin, F., Dance, C. Fisher kernels on visual vocabularies for image categorization. *CVPR*; 2007.
31. Krapac, J., Verbeek, J., Jurie, F. Modeling spatial layout with fisher vectors for image categorization. *ICCV*; 2011.
32. Sydorov, V., Sakurada, M., Lampert, C. Deep fisher kernels - end to end learning of the fisher kernel gmm parameters. *CVPR*; 2014.
33. Maaten, L. Learning discriminative fisher kernels. *ICML*; 2011. p. 217-224.
34. Chapelle O, Vapnik V, Bousquet O, Mukherjee S. Choosing multiple parameters for support vector machines. *Machine Learning*. 2002; 46(1-3):131–159.
35. Wang L. Feature selection with kernel class separability. *IEEE TPAMI*. 2008; 30(9):1534–1546.
36. Liu, X., Wang, L., Yin, J., Zhu, E., Zhang, J. An efficient approach to integrating radius information into multiple kernel learning. *IEEE. TSMC-B*; 2012.
37. Bishop, C. *Pattern Recognition and Machine Learning*. Springer; 2007.
38. Peharz, R., Pernkopf, F. Exact maximum margin structure learning of bayesian networks. *ICML*; 2012.
39. Zou H. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*. 2006; 101(476):1418–1429.
40. Shojaie A, Michailidis G. Penalized likelihood methods for estimation of sparse high dimensional directed acyclic graphs. *Biometrika*. 2010; 97(3):519–538. [PubMed: 22434937]
41. Fu F, Zhou Q. Learning sparse causal gaussian networks with experimental intervention: Regularization and coordinate descent. *Journal of the American Statistical Association*. 2013; 108(501):288–300.
42. ADNI. <http://www.adni-info.org>
43. Kabani N, MacDonald J, Holmes C, Evans A. A 3d atlas of the human brain. *Neuroimage*. 1998; 7:S7–S17.
44. Tijms B, Seris P, Willshaw D, Lawrie S. Similarity-based extraction of individual networks from gray matter mri scans. *Cereb Cortex*. 2012; 22(7):1530–1541. [PubMed: 21878484]
45. <http://www.cs.huji.ac.il/site/labs/compbio/Repository/>.
46. Mackey, D. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press; 2003.
47. Margaritis, D., Thrun, S. Bayesian network induction via local neighborhoods. *Proceedings of NIPS*; 1999.
48. Tsamardinos, I., Aliferis, C. Towards principled feature selection: Relevancy, filters and wrappers. *International Workshop on Artificial Intelligence and Statistics*; 2003.
49. Spirtes, P., Glymour, C., Scheines, R. *Causation, Prediction, and Search*. Springer-Verlag; 1993.
50. Supekar K, Menon V, Rubin D, Musen M, Greicius M. Network analysis of intrinsic functional brain connectivity in alzheimer’s disease. *PLoS Computational Biology*. 2008; 4(6):1–11.

51. Wang K, Liang M, Wang L, Tian L, Zhang X, Li K, Jiang T. Altered functional connectivity in early alzheimer's disease: a resting-state fmri study. *Human Brain Mapping*. 2007; 28(10):967–978. [PubMed: 17133390]
52. Gould R, Arroyo B, Brown R, Owen A, Bullmore E, Howard R. Brain mechanisms of successful compensation during learning in alzheimer disease. *Neurology*. 2006; 67(6):1011–1017. [PubMed: 17000970]

Appendix

Proposition 1

Given a sparse Gaussian Bayesian Network parameterized by Θ and its associated directed graph \mathcal{G} with m nodes, the graph \mathcal{G} is DAG if and only if there exist some $o_i (i = 1, \dots, m)$ and $\Upsilon \in \mathbb{R}^{m \times m}$, such that for arbitrary $\Delta > 0$, the following constraints are satisfied:

$$o_j - o_i \geq \frac{\Delta}{m} - \Upsilon_{ij}, \quad \forall i, j \in \{1, \dots, m\}, \quad i \neq j \quad (1a)$$

$$\Upsilon_{ij} \geq 0, \quad (1b)$$

$$\Upsilon_{ij} \times \Theta_{ij} = 0, \quad (1c)$$

$$\Delta \geq o_i \geq 0. \quad (1d)$$

Proof

As is known, a Bayesian network is equivalent to a topological ordering (Chapter 8, Section 8.1 on Page 362 in [37]). Therefore, we prove Proposition 1 by showing that i) Eqn. (1a ~ 1d) lead to a topological ordering (the necessary condition), and ii) a topological ordering from a DAG can meet the requirements in Eqn. (1a ~ 1d) (sufficient condition).

First, we prove the necessary condition by contradiction (Fig. 6). We consider three cases for two nodes j and i . Case 1) the nodes j and i are directly connected. If there is an edge from node i to node j , the parameter Θ_{ij} is then non-zero, and thus Υ_{ij} must be zero. According to Eqn. (1a), we have $o_j > o_i$. If at the same time, there is an edge from node j to node i , similarly we have $o_i > o_j$, which contradicts with $o_j > o_i$, and therefore is impossible. Case 2: the nodes j and i are not directly linked but connected by a path. Suppose there is a directed path $P1$ from node i to node j , where $P1$ is composed of nodes k_1, k_2, \dots, k_{m1} in order. Following the above proof, we can have $o_j > o_{k_{m1}} > \dots > o_{k_1} > o_i$. If at the same time another directed path $P2$ links node j to node i , where $P2$ is composed of nodes l_1, l_2, \dots, l_{m2} in order, similarly we have $o_i > o_{l_{m2}} > \dots > o_{l_1} > o_j$, making the contradiction. Case 3) If there is no edge between node i and node j , by definition $\Theta_{ij} = 0$. It is straightforward to see

Eqn. (1b) and Eqn. (1c) hold for any arbitrary non-negative Υ_{ij} . Moreover, for any o_i and o_j satisfying Eqn. (1d), we can show that as long as $\Upsilon_{ij} \geq (\frac{1}{m}+1)\Delta$ (which is positive), Eqn. (1a) will always hold. This is further explained as follows. By Eqn. (1d), we have $o_j - o_i \geq -\Delta \geq \frac{\Delta}{m} - \Upsilon_{ij}$. For Eqn. (1a) to be always held, we need some Υ_{ij} such that $o_j - o_i \geq -\Delta \geq \frac{\Delta}{m} - \Upsilon_{ij}$, which requires $\Upsilon_{ij} \geq (\frac{1}{m}+1)\Delta$. Therefore, there exist a set of o_i and Υ valid for Eqn. (1a ~ 1d) when no edge links node i and node j . In sum, Eqn. (1a ~ 1d) show a topological ordering, that is, if node j comes after node i (that is, $o_j > o_i$) in the ordering, there can not be a link from node j to node i , which guarantees the acyclicity.

Now let us consider the sufficient condition. if \mathcal{G} is a DAG, we can obtain some topological ordering $(1, 2, \dots, m)$ from it. Let \tilde{o}_i be the index of node i in this ordering. Setting $o_i = (\tilde{o}_i - 1) \frac{\Delta}{m}$ ($\forall i \in \{1, \dots, m\}$) we have $\min(o_i) = (1-1) \frac{\Delta}{m} = 0$ and $\max(o_i) = (m-1) \frac{\Delta}{m} \leq \Delta$. If node j comes after node i , we have $o_j - o_i \geq \frac{\Delta}{m} \geq \frac{\Delta}{m} - \Upsilon_{ij}$. If node j comes before node i , we can always set Υ_{ij} sufficiently large to satisfy Eqn. (1a ~ 1d). Therefore, from a DAG, we can always construct a set of ordering variables that satisfy Eqn. (1a ~ 1d).

Combining the proofs above, Eqn. (1a ~ 1d) are the sufficient and necessary condition for a directed graph \mathcal{G} to be DAG.

Proposition 2

The optimization problem in Eqn. (2) (i.e., Eqn. (4.2) in the paper) is iteratively solved by alternate optimizations of (i) \mathbf{o} and \mathbf{T} with Θ fixed, and (ii) Θ with \mathbf{o} and \mathbf{T} fixed. This optimization converges and the output Θ^\star is DAG when $\lambda_{dag} > \frac{2m(m-2)(n-1)^2 + m\lambda_1(2n-2-\lambda_1)}{\lambda_1(1+m)\Delta}$, where m is the number of nodes and n is the number of samples.

$$\begin{aligned} \min_{\Theta, \mathbf{o}, \Upsilon} \sum_{i=1}^m \|\mathbf{x}_{:,i} - \mathbf{P}\mathbf{A}_i^\top \boldsymbol{\theta}_i\|_2^2 + \lambda_1 \|\boldsymbol{\theta}_i\|_1 + \lambda_{dag} \varepsilon_i^\top |\boldsymbol{\theta}_i| \\ \text{s.t. } o_j - o_i \geq \frac{\Delta}{m} - \Upsilon_{ij}, \forall i, j \in \{1, \dots, m\}, i \neq j \\ 0 \leq o_i \leq \Delta, \Upsilon_{ij} \geq 0 \end{aligned} \quad (2)$$

Here \mathbf{o} and \mathbf{T} are the variables defined in the DAG constraint in Section 4.2, and Θ is the model parameters of SGBN. The vector \mathbf{e}_i denotes the i -th column of the matrix \mathbf{T} , and $|\boldsymbol{\theta}_i|$ the component-wise absolute value of the i -th column of Θ . Other parameters are defined in Table 1 in the paper.

Proof

In the following, we prove that:

1. The alternate optimization in Eqn. (2) converges.
2. The solution Θ^\star of Eqn. (2) is DAG when λ_{dag} is sufficiently large.

Let us denote $f(\Theta, \mathbf{o}, \Upsilon) = \sum_{i=1}^m \|\mathbf{x}_{:,i} - \mathbf{P}\mathbf{A}_i^\top \boldsymbol{\theta}_i\|_2^2 + \lambda_1 \|\boldsymbol{\theta}_i\|_1 + \lambda_{dag} \varepsilon_i^\top |\boldsymbol{\theta}_i|$.

First, we prove Eqn. (2) converges by showing that (i) $f(\Theta, \mathbf{o}, \Upsilon)$ is lower bounded; and (ii) $f(\Theta^{(t+1)}, \mathbf{o}^{(t+1)}, \Upsilon^{(t+1)}) \leq f(\Theta^{(t)}, \mathbf{o}^{(t)}, \Upsilon^{(t)})$, meaning that the function value will monotonically decrease with the iteration number t .

It is easy to see that $f(\Theta, \mathbf{o}, \Upsilon)$ is lower bounded by 0, since each term in $f(\Theta, \mathbf{o}, \Upsilon)$ is non-negative. And the second point can be proven as follows. At the t -th iteration, we update Θ by

$$\begin{aligned} \Theta^{(t+1)} &= \arg \min_{\Theta} \sum_{i=1}^m \|\mathbf{x}_{:,i} - \mathbf{P}\mathbf{A}_i^\top \boldsymbol{\theta}_i\|_2^2 + \lambda_1 \|\boldsymbol{\theta}_i\|_1 + \lambda_{dag} \varepsilon_i^{(t)\top} |\boldsymbol{\theta}_i| \\ &= \arg \min_{\Theta} f(\Theta, \mathbf{o}^{(t)}, \Upsilon^{(t)}). \end{aligned} \quad (3)$$

It holds that $f(\Theta^{(t+1)}, \mathbf{o}^{(t)}, \Upsilon^{(t)}) \leq f(\Theta^{(t)}, \mathbf{o}^{(t)}, \Upsilon^{(t)})$. Also it is noted that $\Theta^{(t+1)}$ is an achievable global minimum of Θ since $f(\Theta, \mathbf{o}^{(t)}, \Upsilon^{(t)})$ is a convex function with respect to Θ . Similarly, we then update \mathbf{o} and Υ by

$$\begin{aligned} \{\mathbf{o}^{(t+1)}, \Upsilon^{(t+1)}\} &= \arg \min_{\mathbf{o}, \Upsilon} f(\Theta^{(t+1)}, \mathbf{o}, \Upsilon) \\ \text{s.t. } o_j - o_i &\geq \frac{\Delta}{m} - \Upsilon_{ij}, \forall i, j \in \{1, \dots, m\}, i \neq j \\ 0 &\leq o_i \leq \Delta, \Upsilon_{ij} \geq 0. \end{aligned} \quad (4)$$

It holds that $f(\Theta^{(t+1)}, \mathbf{o}^{(t+1)}, \Upsilon^{(t+1)}) \leq f(\Theta^{(t+1)}, \mathbf{o}^{(t)}, \Upsilon^{(t)})$. Also, $f(\Theta^{(t+1)}, \mathbf{o}, \Upsilon)$ is a linear function with respect to \mathbf{o} and Υ . Consequently we have

$$f(\Theta^{(t+1)}, \mathbf{o}^{(t+1)}, \Upsilon^{(t+1)}) \leq f(\Theta^{(t+1)}, \mathbf{o}^{(t)}, \Upsilon^{(t)}) \leq f(\Theta^{(t)}, \mathbf{o}^{(t)}, \Upsilon^{(t)}).$$

Therefore, the optimization problem in Eqn. (2) is guaranteed to converge with the alternate optimization strategy, because the objective function is lower-bounded and monotonically decreases with the iteration numbers.

Second, we prove that when $\lambda_{dag} > \frac{2m(m-2)(n-1)^2 + m\lambda_1(2n-2-\lambda_1)}{\lambda_1(1+m)\Delta}$, the output Θ^* is guaranteed to be DAG. This could be proven by contradiction. Suppose that such a λ_{dag} does not lead to a DAG, say, $\Upsilon_{ji} \times \Theta_{ji}^* \neq 0$ for at least one pair of nodes i and j , with $\Theta_{ji}^* \neq 0$ and $\Upsilon_{ji} > 0$.

Without loss of generality, we assume $\Upsilon_{ji} \geq (\frac{1}{m} + 1)\Delta$ (where $\frac{1}{m}$ is an arbitrary positive number), so the ordering constraints in Eqn. (2) always hold regardless of the variables o_i and o_j . This is because o_i and o_j are constrained by $0 \leq o_i \leq \Delta$ and $0 \leq o_j \leq \Delta$, and

$o_j - o_i \geq -\Delta = \frac{1}{m}\Delta - (\frac{1}{m} + 1)\Delta$. Based on the first-order optimality condition, $\Theta_{ji}^* \neq 0$ i.f.f.

$$2 \left| \left(\mathbf{x}_{:,i} - \mathbf{P}\mathbf{A}_{i(\setminus j,:)}^\top \boldsymbol{\theta}_{i \setminus j}^* \right)^\top \mathbf{x}_{:,j} \right| - (\lambda_1 + \lambda_{dag} \Upsilon_{ij}) > 0. \text{ Here, } \mathbf{P}\mathbf{A}_{i(\setminus j,:)} \text{ denotes the elements in the matrix } \mathbf{P}\mathbf{A}_i \text{ with the } j\text{-th row removed (i.e., parents of the node } i \text{ without the node } j), \text{ and } \boldsymbol{\theta}_{i \setminus j}^* \text{ denotes the elements in } \boldsymbol{\theta}_i^* \text{ without } \boldsymbol{\theta}_{ji}^*.$$

However, it can be shown that,

$$\begin{aligned}
\left| \left(\mathbf{x}_{:,i} - \mathbf{P} \mathbf{A}_{i(\setminus j,:)}^\top \boldsymbol{\theta}_{i \setminus j}^* \right)^\top \mathbf{x}_{:,j} \right| &\leq |\mathbf{x}_{:,i}^\top \mathbf{x}_{:,j}| + \left| \boldsymbol{\theta}_{i \setminus j}^{*\top} \mathbf{P} \mathbf{A}_{i(\setminus j,:)} \mathbf{x}_{:,j} \right| \\
&= |\mathbf{x}_{:,i}^\top \mathbf{x}_{:,j}| + \sum_{k=1, k \neq i, j}^m |\Theta_{ki}^* \mathbf{x}_{:,k}^\top \mathbf{x}_{:,j}| \\
&\leq (n-1) + (m-2)(n-1) \max |\Theta_{ki}^*| \\
&\leq (n-1) + \frac{(m-2)(n-1)^2}{\lambda_1}.
\end{aligned} \tag{5}$$

The second last inequality holds due to the normalization of features $\mathbf{x}_{:,j}$ (to zero mean and unit std). The last inequality holds because

$$\begin{aligned}
\max |\Theta_{ki}^*| &\leq \|\boldsymbol{\theta}_i^*\|_1 \leq \frac{1}{\lambda_1} \left(\|\mathbf{x}_{:,i} - \mathbf{P} \mathbf{A}_i^\top \boldsymbol{\theta}_i^*\|_2^2 + \lambda_1 \|\boldsymbol{\theta}_i^*\|_1 + \lambda_{dag} \boldsymbol{\varepsilon}_i^{*\top} |\boldsymbol{\theta}_i^*| \right) \\
&= \frac{1}{\lambda_1} f(\boldsymbol{\Theta}^*, \mathbf{o}^*, \boldsymbol{\Upsilon}^*) \leq \frac{1}{\lambda_1} f(\mathbf{0}, \mathbf{o}^*, \boldsymbol{\Upsilon}^*) \\
&= \frac{1}{\lambda_1} \mathbf{x}_{:,i}^\top \mathbf{x}_{:,i} = \frac{n-1}{\lambda_1}.
\end{aligned}$$

Eqn. (5) results in . With the given λ_{dag} ,

$$2 \left| \left(\mathbf{x}_{:,i} - \mathbf{P} \mathbf{A}_{i(\setminus j,:)}^\top \boldsymbol{\theta}_{i \setminus j}^* \right)^\top \mathbf{x}_{:,j} \right| - (\lambda_1 + \lambda_{dag} \boldsymbol{\Upsilon}_{ij}) < 0,$$

which contradicts the above first-order optimality condition with $\boldsymbol{\Theta}_{ji}^* \neq 0$. Therefore, when λ_{dag} is sufficiently large, the output $\boldsymbol{\Theta}^*$ is guaranteed to be DAG.

Summing up the proofs above, the alternate optimization of Eqn. (2) converges and the output $\boldsymbol{\Theta}^*$ is guaranteed to be DAG when λ_{dag} is sufficiently large.

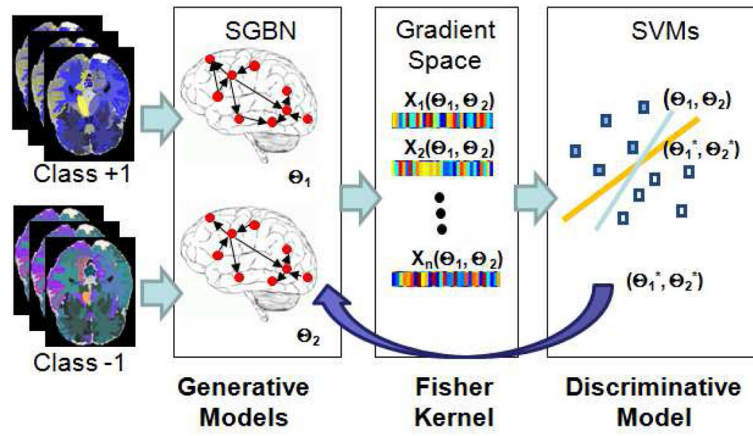


Fig. 1.
Illustration of Fisher-kernel-induced Discriminative Learning.

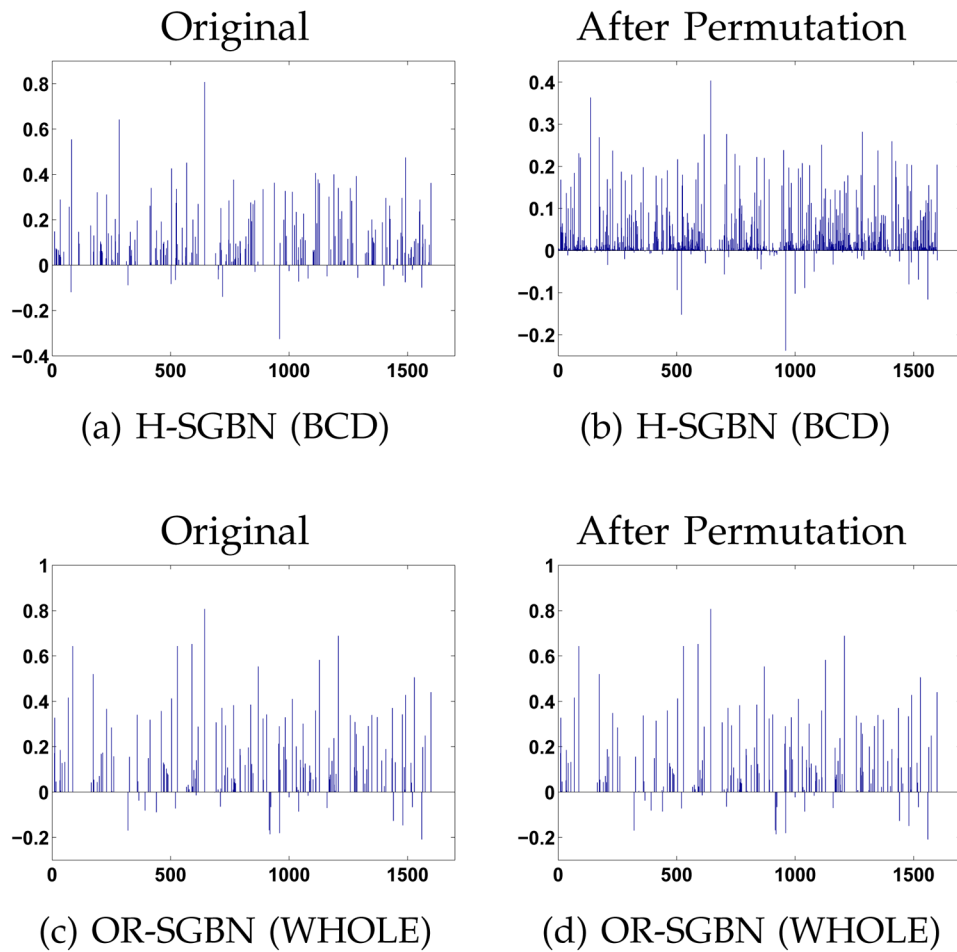


Fig. 2. One example of the estimated parameter Θ for the MCI class (reshaped as a long vector) with regard to the random permutation of the feature ordering. Quantitative measurements of the changes are given in Table 3.

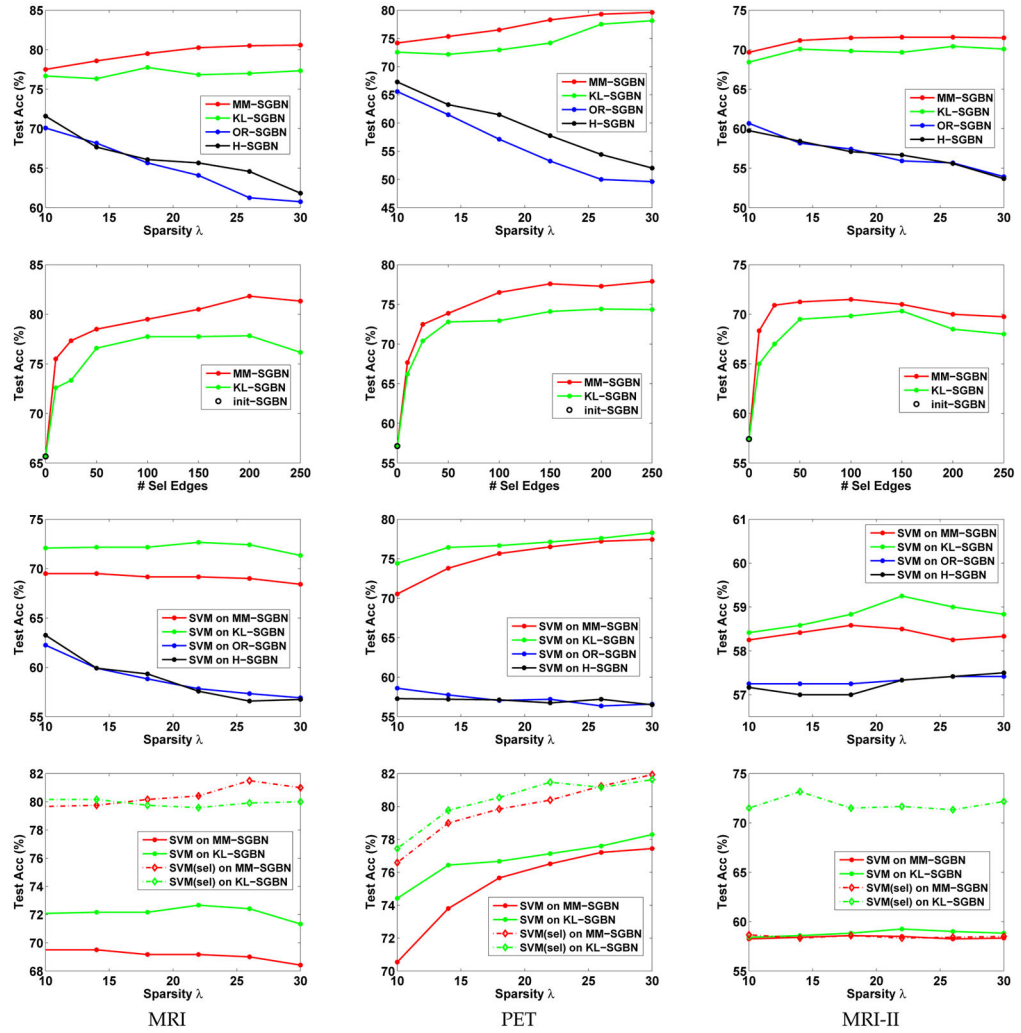


Fig. 3. Comparison of classification accuracies on data sets of MRI (the left column), PET (the middle column) and MRI-II (the right column). The top two rows correspond to the test accuracies obtained by the learned SGBNs. The first row shows the test accuracies varied with the sparsity levels (i.e., the parameter λ). The second row shows the test accuracies varied with the number of edges (denoted as “#Sel Edges” in the figure) optimized in discriminative learning. The bottom two rows correspond to the test accuracies obtained by SVMs using the SGBN-induced Fisher vectors either in full length (the third row) or with (100) selected components (the fourth row).

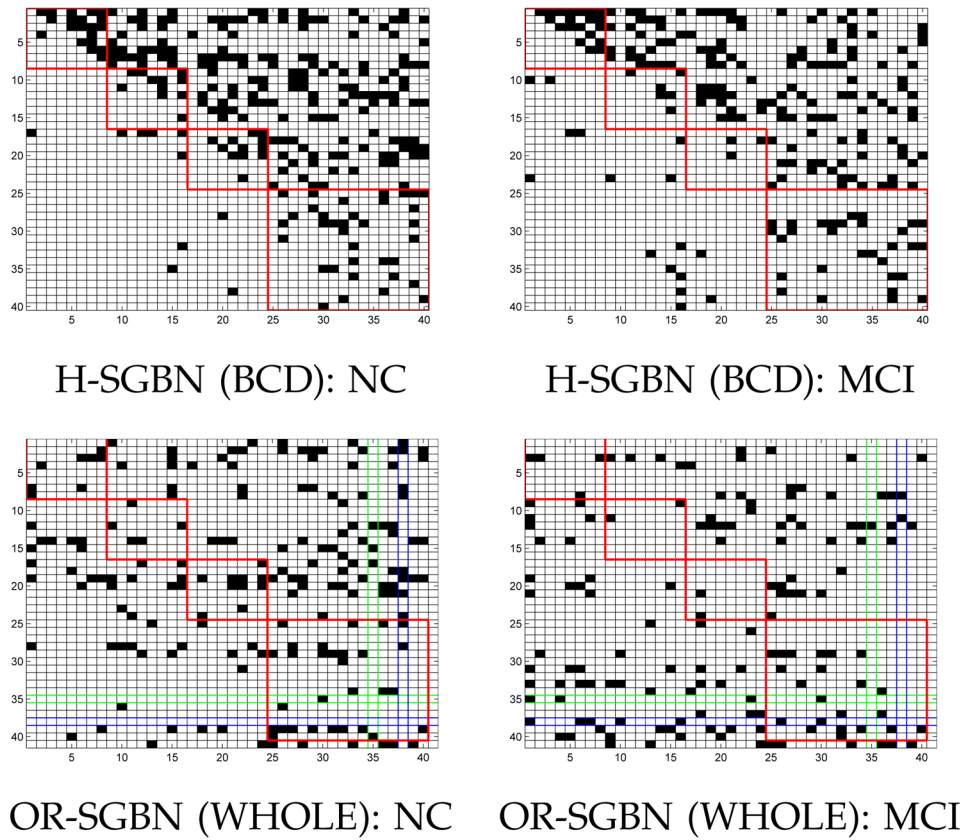


Fig. 4. Visualization of connectivities for MRI-II. The four red boxes correspond to the frontal, parietal, occipital and temporal (including subcortical regions) lobes of the brain. The green row (Row 35) and column (Col 35) correspond to the left hippocampus while the blue ones (Row 38 and Col 38) correspond to the right hippocampus.

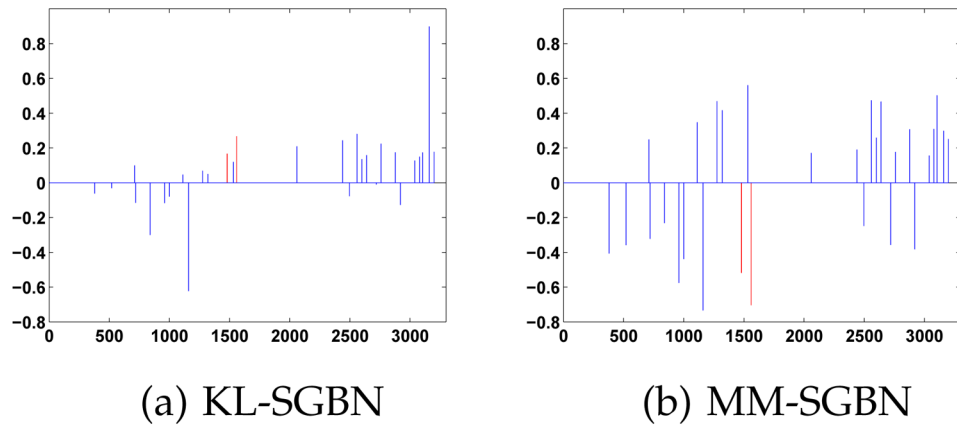


Fig. 5.
An example: change of edge weights learned by KL-SGBN and MM-SGBN

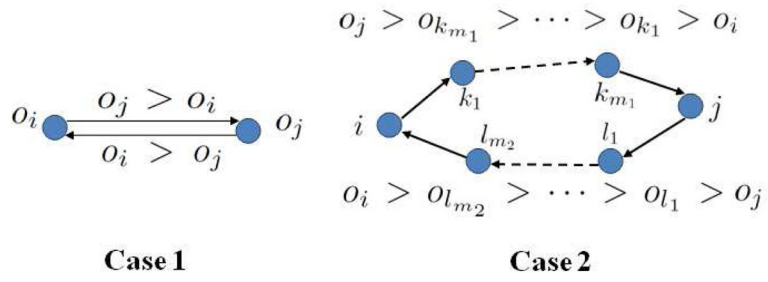


Fig. 6.
 Explanation of our ordering based DAG constraint.

TABLE 1

Notation

x_j	a random variable
\mathbf{x}	a sample of m variables: $\mathbf{x} = [x_1, x_2, \dots, x_m]^\top$
\mathbf{X}	the data matrix of n samples, $\mathbf{X} \in \mathbb{R}^{n \times m}$
\mathbf{x}_i	the i -th row of \mathbf{X} , representing a sample
$\mathbf{x}_{:j}$	the j -th column of \mathbf{X} , representing the realization of the random variable x_j on n samples
$\boldsymbol{\theta}$	the parameters of a Gaussian Bayesian Network $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_m]$, $\boldsymbol{\theta} \in \mathbb{R}^{m \times m}$
\mathbf{Pa}_j	a vector containing the parents of x_j
\mathbf{PA}_j	a matrix whose j -th column represents a realization of \mathbf{Pa}_j on the j -th sample.
\mathbf{G}	an $m \times m$ matrix for BN: if there is a directed <i>edge</i> from x_i to x_j , $\mathbf{G}_{ij} = 1$, otherwise $\mathbf{G}_{ij} = 0$
\mathbf{P}	an $m \times m$ matrix for BN: if there is a directed <i>path</i> from x_i to x_j , $\mathbf{P}_{ij} = 1$, otherwise $\mathbf{P}_{ij} = 0$

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

TABLE 2

Summary of Experiment Purpose

Experiment	Test Subject	Purpose
Exp-I (Sec. 5.2)	DAG constraint	Test the invariance of solution to feature ordering
Exp-II (Sec. 5.2)	DAG constraint	Test the ability of network structure recovery
Exp-III (Sec. 5.3)	discriminative learning	Test the improvement of discriminative power of SGBN models
Exp-IV (Sec. 5.4)	brain network analysis	Investigate the learned brain connectivity patterns

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

TABLE 3

Quantitative Analysis of Θ for the random permutation of feature ordering (between the original and the averaged Θ)

		Distance	Correlation (R)
OR-SGBN (WHOLE)	Θ_1	0.08	0.9996
	Θ_2	0.18	0.9981
H-SGBN (BCD)	Θ_1	1.91	0.6828
	Θ_2	2.06	0.6396

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Total errors (number of both false and missing edges, averaged on 50 simulations) on benchmark networks

TABLE 4

	LIMB	GS	TC-bw	TC	IAMB	IAMB1	IAMB2	IAMB3	OR-SGBN
Factors	101.48	104.50	102.90	103.02	103.14	103.30	103.14	103.14	54.82
Alarm	56.58	59.30	57.76	60.60	61.76	59.16	61.76	61.76	44.40
Barley	113.24	114.70	114.38	122.78	123.80	109.92	123.80	123.80	99.26
Carpo	125.74	131.72	131.18	133.16	132.76	132.90	132.76	132.76	25.58
Chain	5.32	4.88	5.50	4.42	4.70	5.00	4.70	4.70	7.04
Hailfinder	91.50	94.94	96.18	99.02	103.10	92.74	103.10	103.10	57.04
Insurance	74.78	74.64	73.74	76.30	78.78	73.04	78.78	78.78	59.04
Mildew	60.86	60.74	59.66	63.80	68.46	92.74	103.10	103.10	57.04
Water	92.86	94.04	90.24	97.16	102.70	90.06	102.70	102.70	93.08

Number of falsely identified edges (averaged on 50 simulations) on benchmark networks

TABLE 5

	LIMB	GS	TC-bw	TC	IAMB	IAMB1	IAMB2	IAMB3	OR-SGBN
Factors	47.66	50.74	49.40	49.74	50.28	49.70	50.28	50.28	17.70
Alarm	36.04	37.72	36.86	39.24	40.96	37.30	40.96	40.96	23.14
Barley	71.70	72.30	72.60	80.76	82.96	69.76	82.96	82.96	48.70
Carpo	71.96	76.30	75.14	77.38	77.18	77.36	77.18	77.18	14.56
Chain	2.66	2.44	2.76	2.22	2.36	2.50	2.36	2.36	3.52
Haifinder	60.00	62.04	63.16	65.42	66.40	60.90	66.40	66.40	28.66
Insurance	42.80	42.16	41.72	44.06	48.08	41.42	48.08	48.08	31.20
Mildew	46.22	46.46	45.46	49.82	52.48	44.82	52.48	52.48	33.86
Water	64.52	65.02	63.70	68.06	74.22	63.48	74.22	74.22	46.74

Number of falsely identified P-DAG structures (averaged on 50 simulations) on benchmark networks

TABLE 6

	LMB	GS	TC-bw	TC	IAMB	IAMB1	IAMB2	IAMB3	OR-SGBN
Factors	107.20	109.54	108.96	108.84	109.22	108.84	109.22	109.22	63.40
Alarm	61.74	64.08	62.54	65.34	66.82	63.98	66.82	66.82	51.02
Barley	120.54	122.26	121.38	130.04	131.24	116.92	131.24	131.24	105.50
Carpo	129.02	135.34	134.78	136.92	136.74	136.22	136.74	136.74	33.74
Chain	5.96	5.54	6.12	5.16	5.30	5.66	5.30	5.30	7.42
Haifinder	103.72	106.08	107.56	110.04	113.44	104.86	113.44	113.44	63.76
Insurance	81.58	81.68	81.44	83.70	86.60	80.66	86.60	86.60	68.26
Mildew	61.68	61.32	60.34	64.48	69.30	58.08	69.30	69.30	67.24
Water	96.34	97.46	93.80	100.38	106.14	93.60	106.14	106.14	94.52