Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 20(10): 17702–17725.
DOI: 10.3934/mbe.2023787
Received: 29 April 2023
Revised: 19 July 2023
Accepted: 16 August 2023
Published: 15 September 2023

Research article

Network-based diagnostic probability estimation from resting-state functional magnetic resonance imaging

Atsushi Kawaguchi*

Faculty of Medicine, Saga University, Japan

* **Correspondence:** Email: akawa@cc.saga-u.ac.jp.

Abstract: Brain functional connectivity is a useful biomarker for diagnosing brain disorders. Connectivity is measured using resting-state functional magnetic resonance imaging (rs-fMRI). Previous studies have used a sequential application of the graphical model for network estimation and machine learning to construct predictive formulas for determining outcomes (e.g., disease or health) from the estimated network. However, the resulting network had limited utility for diagnosis because it was estimated independent of the outcome. In this study, we proposed a regression method with scores from rs-fMRI based on supervised sparse hierarchical components analysis (SSHCA). SSHCA has a hierarchical structure that consists of a network model (block scores at the individual level) and a scoring model (super scores at the population level). A regression model, such as the multiple logistic regression model with super scores as the predictor, was used to estimate diagnostic probabilities. An advantage of the proposed method was that the outcome-related (supervised) network connections and multiple scores corresponding to the sub-network estimation were helpful for interpreting the results. Our results in the simulation study and application to real data show that it is possible to predict diseases with high accuracy using the constructed model.

Keywords: Alzheimer's disease; brain network; supervised sparse hierarchical component analysis; scoring; dimension reduction

1. Introduction

The evaluation of brain functional networks and network connectivity is an important approach for the study of brain disorders [1]. Functional connectivity is measured using resting-state functional magnetic resonance imaging (rs-fMRI), which has a spatial-temporal 4-dimensional data structure with a voxel size of $64 \times 64 \times 49$ for spatial and 128 time points in typical cases. Different patterns of functional connectivity can be used as biomarkers for disease diagnosis and assessment [2]. Recent studies have employed statistical models and machine learning techniques to assess individual

networks of anatomical brain areas in the context of several brain disorders, such as Alzheimer's disease (AD) [3], mild cognitive impairment (MCI) [4], a review of research on AD and MCI [5, 6], autism spectrum disorder [7], schizophrenia [8], major depressive disorder [9], chronic insomnia disorder [10], and attention deficit hyperactivity disorder [11].

Prediction models have been an important topic in current research [12, 13]. [14] established a data-driven framework of connectome-based predictive modeling, which was utilized in the protocol proposed by [15]. In recent studies related to the framework of connectome-based modeling, many found it useful to use neural networks. In [16], a deep 3D convolutional neural network (3DCNN) was trained on a large cohort of healthy subjects of a wide age range to produce a map representing the probability that a voxel belongs to a particular brain network. [17] developed an attention-based graphical neural network (GNN) framework to detect accelerated brain aging in AD patients. First, graph data were constructed from Pearson correlation matrices computed from rs-fMRI, and then GNN models were trained using the graph data to predict the brain age of HC, MCI patients and AD patients. Although there have been many studies on rs-fMRI, our study focuses on a different target, diagnostic aids and proposes a different analysis approach. In general, the input for predictive modeling is not the observed data itself, but the features, whose construction and selection is an important process in the analysis. There are several types of features, such as a mean time series of the regions of interest (ROI) [18], a graph of theoretical indicators such as small-worldness [19, 20], and connectivity strength (edges) estimated from the mean time series of the ROI. In this study, edges were targeted for ease of interpretation and usefulness of prediction because of direct output from the networks. Several approaches have been used to estimate brain networks from rs-fMRI data, including pairwise Pearson's correlation analyses [21, 22], partial correlation analyses [23, 24], independent component analysis [25] and sparse regressions [26]. Partial correlation coefficients are easily implemented and have been, often selected for this type of analysis. The graphical least absolute shrinkage and selection operator (glasso) [27, 28] provided the sparse estimation of the partial correlation coefficients. This method is based on the inverse covariance matrix and allows for the choice of connections between regions and is useful for computational cost and interpretation of results.

Previous studies have employed the sequential application of outcome-independent network scores and machine learning to construct a prediction formula for outcomes (e.g., disease or healthy) from the estimated network. In the first step, the average time series within the ROI was computed from a multiple voxel time series within an anatomically defined region, for example, by anatomical automatic labeling. However, this approach provided a less informative network for diagnosis because the network was estimated independent of the outcome. Thus, we consider the sequential approach, in which network estimation is followed by regression analysis, to be potentially inconvenient. Such network estimation methods describe the relationships between brain regions based on a class of correlation coefficients, which do not necessarily include the relationship with the disease. This is unlikely to lead directly to increased accuracy in the diagnostic or predictive models targeted by this study. If the network estimation includes not only the association between brain regions but also the association with disease, the accuracy of diagnosis or prediction could be improved.

We aimed to develop a supervised network estimation method for use in a prediction formula. Additionally, we compared the proposed method with the existing sequential approach. Network estimation is performed by extending the supervised scoring method proposed by [29]. While [29] uses simple linear combination scoring, network estimation in this framework requires hierarchical

(multiple levels) scoring. At the lower level, network estimation is performed for each individual. In a data matrix consisting of multiple node time series (time points in rows, brain region node in columns), the regression model takes the equation form with one node as the objective variable (output values) and the remaining node time series as the explanatory variables (input values). The scores obtained in the lower level are further scored as a subject group in the upper level with information about the diagnosis given as a supervisor. Performing these processes in a single algorithm, it is expected that the network score would also include information about the outcome, and the resulting scores would be useful in improving the accuracy of the diagnostic or predictive model.

In this paper, the scoring methods and algorithms are introduced in Section 2.1. The evaluation of the diagnostic or predictive model accuracy is then planned in Sections 2.2 or 2.3, and the results are presented in Sections 3.1 or 3.2 for the simulation study and the real data application.

2. Methods

In the regression model of this study, the outcome was the response and the regional time series from rs-fMRI was the predictor. We proposed a regression method with rs-fMRI scores based on supervised sparse hierarchical components analysis (SSHCA). The SSHCA had a hierarchical structure consisting of a network model (block scores at the individual level) and a scoring model (super scores at the population level). Multiple super scores (components) were subsequently computed. A regression model with super scores as predictors was used to estimate the diagnostic probability. The methods in this study were implemented in the R programming environment using the latest version of the msma package. The mand package [30] was used to handle the display and other aspects of the brain imaging data. The proposed method was compared with existing methods through simulation studies, and its usefulness was investigated through real data analysis.

2.1. SSHCA

This section describes the score structure and the estimation method of the weights. For the estimation method, we first define the objective function and introduce the algorithm for obtaining its solution. The reasonableness of the algorithm is provided in the Appendix. Notation is given for all beginnings. Considering n subjects, $\boldsymbol{X}_i = (\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \ldots, \boldsymbol{x}_{i,M})$ is the $T \times M$ average time series of ROI ($i = 1, 2, \ldots, n$) and $\boldsymbol{x}_{i,m} = (x_{i,m}(1), x_{i,m}(2), \ldots, x_{i,m}(T))^\top$. T is the number of time points, and M is the number of nodes (ROIs). Subjects also have a univariate outcome, and the n-dimensional outcome vector is denoted by $\boldsymbol{Z}$.

Our basic model was a hierarchical (multiblock) score structure divided into two parts: a population level and an individual level. The individual level could be further divided into two levels, individual bottom and individual top. The network was estimated at the individual bottom level, and the resulting scores were obtained at higher levels. We formulated the following score representation. First, consider the population level score $\boldsymbol{s}$ with the following multiblock (hierarchical) structures:

$$\boldsymbol{s} = \boldsymbol{S}_2 \boldsymbol{w}_2 = \sum_{m=1}^{M} \boldsymbol{s}_{2,m} w_{2,m}$$

where $\boldsymbol{w}_2 = (w_{2,1}, \ldots, w_{2,M})^\top$ is the weight vector with length M for $n \times M$ matrix $\boldsymbol{S}_2$ with the m-th

column $\boldsymbol{s}_{2,m}$ and the i-th (individual level) element is given as

$$s_{2,i,m} = \sum_{t=1}^{T} s_{3,i,m}(t) w_{3,i}(t) = \boldsymbol{s}_{3,i,m}^{\top} \boldsymbol{w}_{3,i}$$

where $\boldsymbol{w}_{3,i} = (w_{3,i}(1), \ldots, w_{3,i}(T))^{\top}$ is the weight vector with length T for the m-th sub-block of i-th subject score $\boldsymbol{s}_{3,i,m}$ given by

$$\boldsymbol{s}_{3,i,m} = \boldsymbol{X}_{i,(-m)} \boldsymbol{w}_{4,i,m}$$

where $\boldsymbol{w}_{4,i,m}$ is the weight vector with length $M - 1$ for the m-th sub-blocks $\boldsymbol{X}_{i,(-m)} = (\boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,m-1}, \boldsymbol{x}_{i,m+1}, \ldots, \boldsymbol{x}_{i,M})$, which is the data matrix $\boldsymbol{X}_i$ except for the m-th column. The t-th element of $\boldsymbol{s}_{3,i,m}$ is also given as $\boldsymbol{s}_{3,i,m}(t) = \boldsymbol{X}_{i,(-m)}^{\top}(t) \boldsymbol{w}_{4,i,m}$.

The optimal value of the weight $\boldsymbol{w}_{4,i,m}$ is obtained by maximizing $\sum_{i=1}^{n} \sum_{m=1}^{M} cov(\boldsymbol{s}_{3,i,m}, \boldsymbol{x}_{im}) = \sum_{i=1}^{n} \sum_{m=1}^{M} cov(\boldsymbol{X}_{i,(-m)} \boldsymbol{w}_{4,i,m}, \boldsymbol{x}_{im})$, which will be discussed in more detail later in the algorithm for finding the weights. This is an original way to create the network developed in this paper. The method is based on a regression model in which one node is removed from a data matrix consisting of a multi-node time series (rows: time points, columns: nodes), and it is used as the objective variable (output values) and the remaining node time series as explanatory variables (input values). It is similar to multiple regression analysis, which analyzes one-to-many relationships between nodes rather than one-to-one relationships like the correlation coefficient. Note that every node is considered to be a node that is the objective variable.

Figure 1 shows a graphical representation of score relationships. The network diagram on the left side of this figure is illustrated using simplified symbols with M=3 nodes. The data matrix can be written as $\boldsymbol{X}_i = (\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \boldsymbol{x}_{i,3})$. The upper network is a model in which ($\boldsymbol{x}_{i,1}$ is the objective variable and the remaining $\boldsymbol{X}_{i,(-1)} = (\boldsymbol{x}_{i,2}, \boldsymbol{x}_{i,3})$ are explanatory variables. The lower network is a model in which ($\boldsymbol{x}_{i,2}$ is the objective variable and the remaining $\boldsymbol{X}_{i,(-2)} = (\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,3})$ are explanatory variables.

Note that the score $\boldsymbol{s}$ could be regarded as population-level scores, and scores $\boldsymbol{s}_{2,i}$ were individual-level scores. There were several types of scores with hierarchical structures, and corresponding weights had the following roles. The weights $\boldsymbol{w}_{4,i,m}$ represented the edge strength to the m-th node variable $\boldsymbol{x}_m$ from the others, and the corresponding score $\boldsymbol{s}_{3im}$ represented the predictor for the node variable $\boldsymbol{x}_m$. The $M \times T$ matrix $\boldsymbol{S}_{3,i}$ consisting of these scores, reduced the time course by using weight $\boldsymbol{w}_{3,i}$. The resulting M-dimensional vector $\boldsymbol{s}_{2,i}$ was used as a representative variable for individual i at the population level; then, at the population level, the score was computed again from these scores by using weight $\boldsymbol{w}_2$. This super score $\boldsymbol{s}$ was used in the prediction model such as the logistic regression model.

In recent years, dynamic network estimation has been widely used in brain image analysis. It is possible to extend the proposed method for such analysis. Because $\boldsymbol{s}_{3,i,m}(t) = \sum_{j \neq m} x_{i,j}(t) w_{4,i,m,j}$, we can rewrite $\boldsymbol{s}_{2,i,m}$ as

$$\boldsymbol{s}_{2,i,m} = \sum_{t=1}^{T} \boldsymbol{s}_{3,i,m}(t) w_{3,i}(t) = \sum_{t=1}^{T} \left\{ \sum_{j \neq m} \boldsymbol{x}_{i,j}(t) w_{4,i,m,j} \right\} w_{3,i}(t)$$

Thus, $\boldsymbol{v}_{i,j,m}(t) = \boldsymbol{w}_{4,i,m,j} w_{3,i}(t)$ could be interpreted as a dynamical relationship between the node j to the node m.

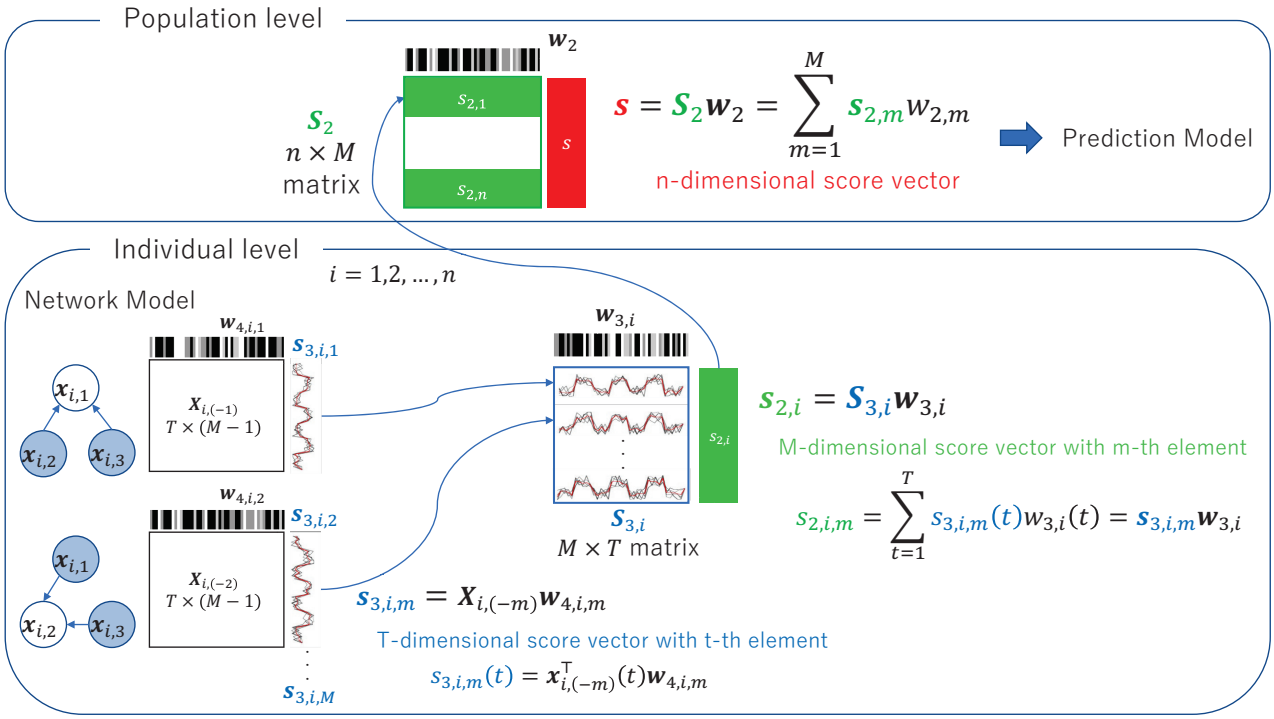

Figure 1. Hierarchical score structure. The hierarchical score structure was divided into two parts: a population level and an individual level. The individual level could be further divided into two levels, individual bottom and individual top.

In the scores given so far, the objective function for the weights to be estimated is given below. An objective function is defined for estimating the network and for summarizing the scores obtained from it and using them for diagnostic probability estimation. This is done in a hierarchical manner, and finally an objective function is considered that optimizes these simultaneously. When matrices $\boldsymbol{X}_i$ were normalized by their column, the weight $\boldsymbol{w} = (\boldsymbol{w}_2^{\top}, \boldsymbol{w}_3^{\top}, \boldsymbol{w}_4^{\top})^{\top}$, where $\boldsymbol{w}_3 = (\boldsymbol{w}_{3,1}^{\top}, \ldots, \boldsymbol{w}_{3,n}^{\top})^{\top}$ and $\boldsymbol{w}_4 = (\boldsymbol{w}_{4,1,1}^{\top}, \ldots, \boldsymbol{w}_{4,n,M}^{\top})^{\top}$ was estimated by maximizing the following function:

$$\begin{aligned} L_0(\boldsymbol{w}) =& \{L_{01}(\boldsymbol{w}_2, \boldsymbol{w}_3, \boldsymbol{w}_4) - P_{2,\lambda_2}(\boldsymbol{w}_2)\} \\ &+ \{L_{02}(\boldsymbol{w}_3, \boldsymbol{w}_4) - P_{3,\lambda_3}(\boldsymbol{w}_3)\} + \{L_{03}(\boldsymbol{w}_4) - P_{4,\lambda_4}(\boldsymbol{w}_4)\}. \end{aligned} \tag{2.1}$$

At the population level, the scores obtained at the individual level are reduced to a single score per person by maximizing the score variance in order to include more information across subjects in the subject population and to reduce the number of nodes dimensionally. In addition to summarizing the individual-level scores in this way at the population level, the correlation between scores and outcomes is also incorporated into the objective function to make the scores useful for the prediction model. From this perspective, the sub objective function for the scoring model in the population level as follows.

$$L_{01}(\boldsymbol{w}_2, \boldsymbol{w}_3, \boldsymbol{w}_4) = (1-\mu) \times var(\boldsymbol{s}) + \mu \times cov(\boldsymbol{s}, \boldsymbol{Z})$$

where $0 \leq \mu \leq 1$ defines the proportion of the supervision. The weights were evaluated by maximizing

the variance of the super scores $\boldsymbol{s}$ supervised by the outcome. In other words, it was obtained by maximizing the variance of the super scores and the covariance with the outcome with a trade-off.

At the individual level, there are two additional layers, with the upper layer summarizing within-subjects the network information obtained in the lower layer. The time series of scores per node obtained in the lower layer is reduced in the time domain by maximizing the score variance and reducing the number of time points, and the score per node is calculated within subjects. The lower layer uses an objective function for network estimation that maximizes the covariance between the linear combined score time series values of one node value and the other node values for each individual. From this perspective, the subobjective functions for the network models in the individual level were as follows.

$$L_{02}(\boldsymbol{w}_3, \boldsymbol{w}_4) = \sum_{i=1}^{n} var(\boldsymbol{s}_{2,i}), \quad L_{03}(\boldsymbol{w}_4) = \sum_{i=1}^{n}\sum_{m=1}^{M} cov(\boldsymbol{s}_{3,i,m}, \boldsymbol{x}_{im})$$

Note that $L_{02}(\boldsymbol{w})$ was a (sub) objective function for the score with temporal (time course) reduction, and $L_{03}(\boldsymbol{w})$ was a (sub) objective function for the score to construct the network.

The function $L_0(\boldsymbol{w})$ was maximized subject to $\|\boldsymbol{w}_2\|^2 = 1$, $\|\boldsymbol{w}_{3,i}\|^2 = 1$ and $\|\boldsymbol{w}_{4,i,m}\|^2 = 1$ ($i = 1, \ldots, n$, $m = 1, \ldots, M$) with parameters regularized to control the sparsity that enabled the detection of associated weights as follows.

$$P_{2,\lambda_2}(\boldsymbol{w}_2) = P_{\lambda_2}(\boldsymbol{w}_2), \quad P_{3,\lambda_3}(\boldsymbol{w}_3) = \sum_{i=1}^{n} P_{\lambda_{3,i}}(\boldsymbol{w}_{3,i}), \quad P_{4,\lambda_4}(\boldsymbol{w}_4) = \sum_{i=1}^{n}\sum_{m=1}^{M} P_{\lambda_{4,i,m}}(\boldsymbol{w}_{4,i,m})$$

where $P_\lambda(x)$ was the penalty function ($P_\lambda(x) = 2\lambda|x|$ in this study), and $\lambda > 0$ the regularized parameter. The function $P_\lambda(x)$ is defined for a scalar input x, but for a vector $\boldsymbol{x}$ it is defined as $P_\lambda(\boldsymbol{x}) = \sum_j P_\lambda(x_j) = 2\lambda \sum_j |x_j|$.

The algorithm for maximizing equation 2.1 is given as follows. The rationale is provided in the Appendix. As defined above, the m-th sub-blocks $\boldsymbol{X}_{i,(-m)}$ is the data matrix $\boldsymbol{X}_i$ except for the m-th column.

1) Initialize $\boldsymbol{s} = (s_1, \ldots, s_n)^\top$, $\boldsymbol{s}_{2,i} = (s_{2,i,1}, \ldots, s_{2,i,M})^\top$, $\hat{\boldsymbol{w}}_2 = (\hat{w}_{2,1}, \hat{w}_{2,2}, \ldots, \hat{w}_{2,M})^\top$ and $\hat{\boldsymbol{w}}_{3,i} = (\hat{w}_{3,i}(1), \hat{w}_{3,i}(2), \ldots, \hat{w}_{3,i}(T))^\top$.
2) Repeat until convergence.
    1) (Individual bottom) $\tilde{\boldsymbol{w}}_{4,i,m} = h_{\lambda_{4,i,m}}(\boldsymbol{X}_{i,(-m)}^\top\{\hat{w}_{2,m}((1-\mu)s_i + \mu Z_i)\hat{\boldsymbol{w}}_{3,i} + \boldsymbol{x}_i\})$ where $h_\lambda(y) = sign(y)(|y| - \lambda)_+$ and normalizes $\hat{\boldsymbol{w}}_{4,i,m} = \tilde{\boldsymbol{w}}_{4,i,m}/\|\tilde{\boldsymbol{w}}_{4,i,m}\|$ ($m = 1, 2, \ldots, M$).
    2) (Individual top) Putting $\boldsymbol{s}_{3,i,m} = \boldsymbol{X}_{i,(-m)}\hat{\boldsymbol{w}}_{4,i,m}$ and $\boldsymbol{S}_{3,i} = (\boldsymbol{s}_{3,i,1}, \ldots, \boldsymbol{s}_{3,i,M})^\top$, $\tilde{\boldsymbol{w}}_{3,i} = \sum_{m=1}^{M}\{\hat{w}_{2,m}((1-\mu)s_i + \mu Z_i)\boldsymbol{s}_{3,i,m} + \boldsymbol{S}_{3,i}^\top\boldsymbol{s}_{2,i}\}$ normalize as $\hat{\boldsymbol{w}}_{3,i} = \tilde{\boldsymbol{w}}_{3,i}/\|\tilde{\boldsymbol{w}}_{3,i}\|$.
    3) (Population) Putting $\boldsymbol{s}_{2,i} = \boldsymbol{S}_{3,i}\hat{\boldsymbol{w}}_{3,i}$, $\tilde{w}_{2,m} = h_{\lambda_{2m}}(\boldsymbol{s}_{2,m}^\top((1-\mu)\boldsymbol{u} + \mu\boldsymbol{Z}))$ then putting $\tilde{\boldsymbol{w}}_2 = (\tilde{w}_{2,1}, \tilde{w}_{2,2}, \ldots, \tilde{w}_{2,M})^\top$ and normalize as $\hat{\boldsymbol{w}}_2 = \tilde{\boldsymbol{w}}_2/\|\tilde{\boldsymbol{w}}_2\|$.
    4) Set $\boldsymbol{s} = \boldsymbol{S}_2\hat{\boldsymbol{w}}_2$.
3) (Deflation step) Set $p_{3,i,m} = \boldsymbol{x}_{i,m}^\top\boldsymbol{s}_{3,i,m}/\boldsymbol{s}_{3,i,m}^\top\boldsymbol{s}_{3,i,m}$ and $\boldsymbol{p}_{3,i} = (p_{3,i,1}, \ldots, p_{3,i,M})$, $\boldsymbol{X}_i$ are deflated by $\boldsymbol{X}_i \leftarrow \boldsymbol{X}_i - \boldsymbol{S}_{3,i}\boldsymbol{p}_{3,i}^\top$ for $i = 1, \ldots, n$. Start again from step 1 and repeat for the given number of times (number of components).

Note that the deflation steps yield multiple components and several alternatives. Extracting multiple components in this way corresponds to multiple estimations of the network, which means that the proposed method can decompose the network. There were several derivations of the parameter update method in Step 2(a). The method that used the update formula written in Step 2(a) was called SSHCA-corde (coordinate updating). Next was the update formula in a form that did not include any weights other than $\boldsymbol{w}_{4,i,m}$, and the method that used $h_{\lambda_{4,i,m}}(\boldsymbol{X}_{i,(-m)}^{\top}\boldsymbol{x}_i)$ as the update formula was called SSHCA-corde.ind (coordinate updating with independent network estimation). The method of estimating $\boldsymbol{w}_{4,i,m}$ using the glasso method was called SSHCA-glasso (coordinate updating with independent glasso network estimation). All these derivations are described in the Appendix and were compared in simulation studies and real data analysis.

The larger value of the regularization parameter λ had many non-zero elements in the weight $\boldsymbol{w}$ values. Its optimal value was selected by minimizing the Bayesian information criterion (BIC). It is denoted by

$$BIC(\lambda) = \log\left(\frac{1}{nTM}\sum_{i=1}^{n}\|\hat{\boldsymbol{X}}_i(\lambda) - \boldsymbol{X}_i\|^2\right) + \frac{\log(nTM)}{nTM}df(\lambda)$$

where $\hat{\boldsymbol{X}}_i(\lambda) = [\hat{\boldsymbol{x}}_{i,1}(\lambda), \ldots, \hat{\boldsymbol{x}}_{i,M}(\lambda)]$, $\hat{\boldsymbol{x}}_{i,m}(\lambda) = \boldsymbol{s}_{3,i,m}p_{3,i,m}^{\top}$. $df(\lambda)$ is the number of effective parameters and depends on the value of λ. In the following, the regularization parameters are simplified such that $\lambda_{3,i} = 0$ and $\lambda_{4,i,m} = \lambda_4$ to avoid redundancy in the calculation.

2.2. Simulation study

The proposed method was evaluated and compared with the sequential approach using synthetic data. The total sample size was $n = 50$ and 100. The true graph had 50 and 100 nodes with edges that were randomly generated with 5 and 20 difference edges between the two groups ($n/2$ sample size per group). Multivariable data with a time length of 100 for the individual were generated as random numbers with a correlation structure using partial correlation coefficients based on the true graph. Then, the actual indicators Z for the case or control were generated by using the binomial random number with the probability being the logistic transformation of the partial correlation coefficients.

The resulting data set was a 100×20 matrix $\boldsymbol{X}_i$ ($i = 1, 2, \ldots, n$). The parameters in the proposed method were set as $\mu = 0$, 0.5 and 1. As explained in the previous section, there were three types of proposed SSHCA methods: SSHCA-corde, SSHCA-corde.ind and SSHCA-glasso. The glasso method was used for network estimation in comparisons. The strength of the edge (penalized estimated partial correlation coefficient) was used as an explanatory variable for prediction using glasso. The machine learning methods: generalized linear model (glm), glmnet, support vector machine (svmRadial), random forests (rf) and neural networks (nnet) were applied for diagnostic probability estimation. This application is also reviewed in the discussion as a consideration. The hyperparameters for machine learning were chosen based on five repeated 10-fold cross-validations. Both estimated networks and diagnostic probabilities were evaluated using receiver operating characteristic (ROC) analysis. In the network estimation, the selected edges were evaluated in the case and control groups. The number of iterations for the above procedures was 50 (the number of simulated data sets).

2.3. Real data analysis

The proposed method was applied to real data from the Alzheimer's Disease Neuroimaging Initiative (http://adni.loni.usc.edu/), a collection of imaging data from 50 subjects at baseline with a mean age of 75 years for 23 healthy subjects and 72.9 years for 27 patients with early MCI (eMCI). Z was a binary variable for Normal, or eMCI. Table 1 summarizes the characteristics of the patients.

Table 1. characteristics for real data

	Normal (n = 23)	eMCI (n = 27)
Age, years	75	72.9
sex, Male [n (%)]	10 (43.5)	17 (63.0)
APOE4≥ 1, [n (%)]	6 (26.1)	14 (51.9)
Mini-Mental Scale Examination score	28.5	27.7

The Data Processing Assistant for Resting-State fMRI (DPARSF) was used to perform rs-fMRI preprocessing, slice timing, realignment, normalization, smoothing, detrending and band path filtering. The resulting data set contained 90 ROIs and 130 time points for each subject. The estimated diagnostic probability was evaluated using ROC analysis. The sensitivity, specificity and area under the curve (AUC) were computed for 20 iterations by taking 70% of the samples randomly, training them and then predicting and evaluating the remaining 30% as a validation set. We compared the proposed method to the existing sequential approach with a glasso as the network estimation, popular machine learning methods (glm, glmnet, svmRadial, rf, nnet as in the simulation study) as the prediction model, and the unsupervised version of our method. The hyperparameters for machine learning were chosen based on five repeated 10-fold cross-validations.

3. Results

3.1. Simulation study

The results in Table 2 are for the following settings: the number of subjects (nsample) is 50, the number of edges (nedge) is 100 and the nedgedif of the edges is 5 and 20. The proportions of nedgedif to nedge were 5 and 20%, respectively. The proposed SSHCA method used four components and the supervision parameters $\mu = 0$, 0.5 and 1. The results for the SSHCA were the area under the ROC curve (pathauc), which is an evaluation index for the true graph structure, the area under the ROC curve (scorecvauc), which is an evaluation index for disease discrimination, and the average of pathauc and scorecvauc (allmean) for each μ. For scorecvauc, the results obtained for each of the five prediction methods glm, glmnet, svmRadial, rf and nnet were averaged.

When μ was changed, the values of pathauc and scorecvauc were both higher when $\mu = 1$. The AUC values were not better when the glasso weights, the method of the previous study, were used directly for prediction. It was no better even in the case of the SSHCA method with $\mu = 1$. The pathauc was higher when the network estimation was done independently (SSHCA-corde.ind or SSHCA-glasso). Furthermore, pathauc was higher when glasso was used for network estimation (SSHCA-glasso), but there was no significant difference when SSHCA-corde.ind was used. Focusing on scorecvauc, the method that performs network and score estimation simultaneously (SSHCA-corde) had the highest

value. However, the scorecvauc of SSHCA-glasso was not comparable, and the average allmean was the largest. SSHCA-corde.ind outperformed SSHCA-glasso in scorecvauc.

Table 2. Simulation study results for *nsample* = 50, *nedge* = 100 and *ncomp* = 4.

nedgedif	Methods	μ	pathauc	scorecvauc	allmean
5(5%)	SSHCA-corde	0.0	0.754	0.523	0.677
		0.5	0.771	0.865	0.802
		1.0	0.781	0.992	0.851
	SSHCA-corde.ind	0.0		0.506	0.721
		0.5	0.829	0.708	0.788
		1.0		0.986	0.881
	SSHCA-glasso	0.0		0.528	0.759
		0.5	0.875	0.757	0.835
		1.0		0.985	0.911
	glasso		0.650	0.547	0.615
20(20%)	SSHCA-corde	0.0	0.748	0.527	0.674
		0.5	0.764	0.870	0.800
		1.0	0.777	0.990	0.848
	SSHCA-corde.ind	0.0		0.531	0.725
		0.5	0.821	0.698	0.780
		1.0		0.984	0.876
	SSHCA-glasso	0.0		0.544	0.747
		0.5	0.847	0.710	0.802
		1.0		0.983	0.893
	glasso		0.649	0.603	0.634

Table 3 shows the results for nsample = 50, nedge = 50 and nedgedif = 5 and 20. The nedge was changed from 100 (in Table 2) to 50. The proportions of nedgedif to nedge were 10 and 40%, respectively.

As the number of nedges decreased, the results were generally better; the percentage of nedgedif was not relevant; the network estimation of glasso was much better, but the predictive power was not very high. The results for the nsample = 100 case are included in the Appendix, but the pattern was the same as for these nsample = 50 cases. In addition, a comparison of the results for each regression model is illustrated in the Appendix. The results show no significant differences among the regression models. The scoring may ensure some high degree of predictive accuracy.

3.2. Real data analysis

We apply the method to real data of AD described in Section 2.3. The network was estimated using three SSHCA methods (corde, corde.ind and glasso) and by the glasso. The results of predicting eMCI are shown in Table 4 as sensitivity, specificity and AUC. We used glm, glmnet, svmRadial, rf and nnet for machine learning, as in the simulation study, and the resulting values are the average among values from those machine learnings.

Table 3. Simulation study results for *nsample* = 50, *nedge* = 50 and *ncomp* = 4.

nedgedif	Methods	μ	pathauc	scorecvauc	allmean
5(10%)	SSHCA-corde	0.0	0.886	0.505	0.759
		0.5	0.895	0.830	0.873
		1.0	0.904	0.987	0.932
	SSHCA-corde.ind	0.0		0.516	0.789
		0.5	0.926	0.713	0.855
		1.0		0.990	0.947
	SSHCA-glasso	0.0		0.522	0.804
		0.5	0.945	0.751	0.880
		1.0		0.984	0.958
	glasso		0.924	0.555	0.801
20(40%)	SSHCA-corde	0.0	0.869	0.551	0.763
		0.5	0.879	0.842	0.867
		1.0	0.887	0.986	0.920
	SSHCA-corde.ind	0.0		0.508	0.742
		0.5	0.859	0.669	0.796
		1.0		0.968	0.895
	SSHCA-glasso	0.0		0.589	0.819
		0.5	0.934	0.767	0.878
		1.0		0.983	0.950
	glasso		0.900	0.686	0.828

Table 4. Real data analysis results.

Methods	μ	Sensitivity	Specificity	AUC
SSHCA-corde	0.00	0.375	0.548	0.607
	0.25	0.583	0.698	0.699
	0.50	0.739	0.796	0.802
	0.75	0.720	0.822	0.847
	1.00	0.713	0.874	0.861
SSHCA-corde.ind	0.00	0.365	0.530	0.633
	0.25	0.408	0.555	0.583
	0.50	0.452	0.598	0.599
	0.75	0.507	0.617	0.621
	1.00	0.753	0.842	0.881
SSHCA-glasso	0.00	0.491	0.614	0.645
	0.25	0.434	0.583	0.579
	0.50	0.455	0.637	0.609
	0.75	0.534	0.654	0.663
	1.00	0.718	0.793	0.826
glasso		0.388	0.556	0.592

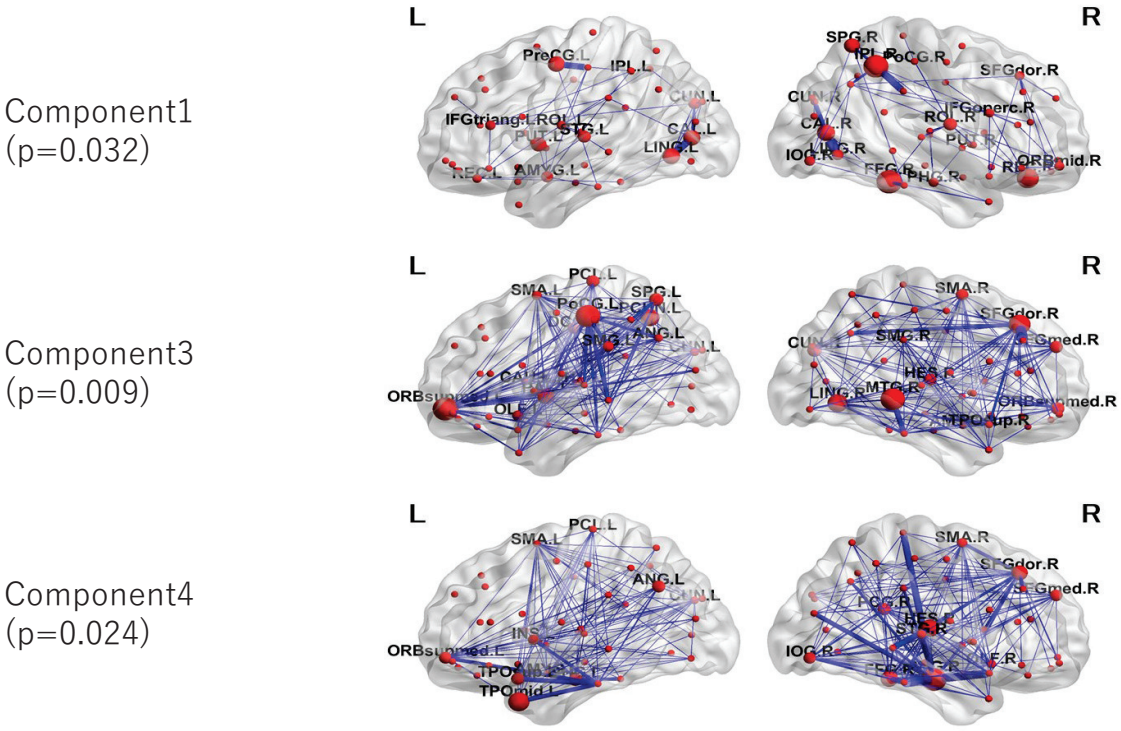

Figure 2. Estimated networks using the SSHCA method where p values were from the logistic regression model.

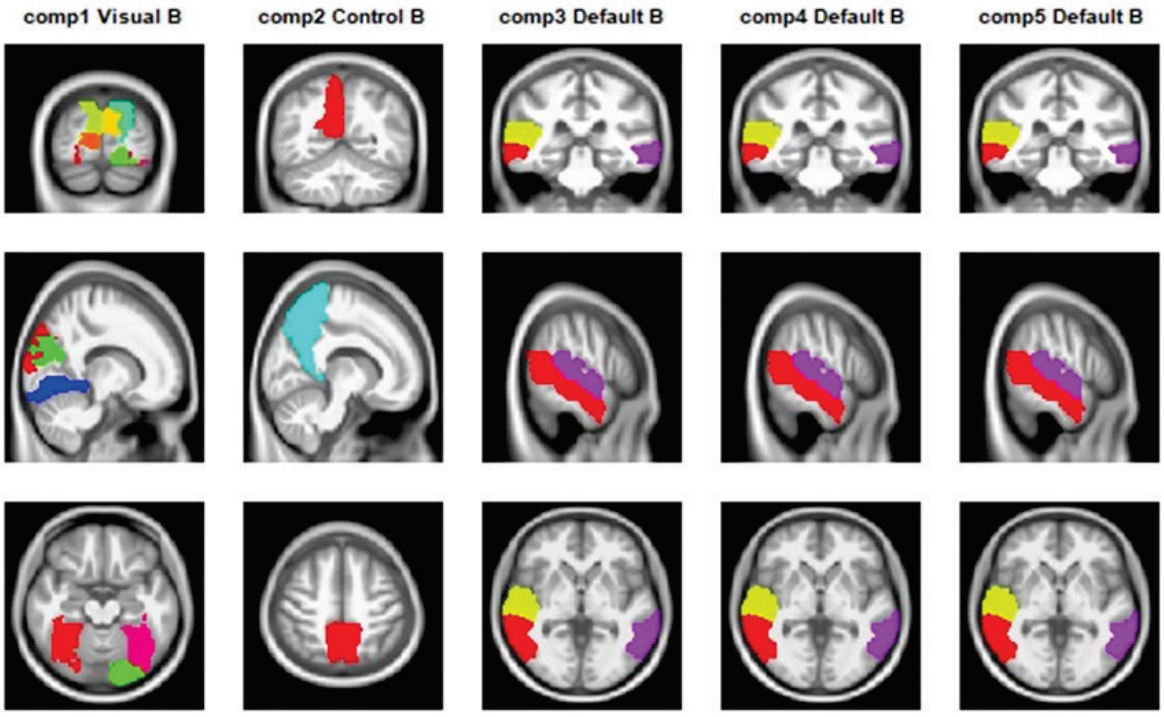

Figure 3. Related reference networks. The most closely related networks estimated for that component are displayed.

The AUC was higher for the proposed SSHCA method than for the glasso method. For SSHCA-corde, the AUC was high, even for $\mu = 0.75$. The highest AUC was 0.881 for SSHCA-corde.ind. Therefore, next we looked closer at the network used in the estimation for the case of SSHCA-corde.ind (μ=1), which had the highest AUC value. The three components were estimated, as shown in Figure 2. The super scores of these networks were significant in the univariate logistic regression model.

We examined which of the networks estimated by each component was closest to the networks examined in previous studies. For the reference network, we used the Yeo_17 network, which is stored in the R package brainGraph and has 17 networks. For each network, we computed the AUC at the edge of the estimated network, and the one with the highest AUC was the one that was most closely related to the network estimated for that component.

The visual network and default mode networks were selected as the significant components in the univariate logistic regression, as shown in Figure 3. As listed in [5], many studies have reported the association between the default mode network and AD, and the results of the present analysis were also reasonable.

4. Discussion

We aimed to characterize brain function based on data measured by fMRI at rest as a time series of voxels arranged in three spatial dimensions and presented a novel regression method based on supervised sparse hierarchical component analysis (SSHCA) with a hierarchical structure consisting of a network model (individual-level block scores) and a scoring model (population-level super score). In addition, the (supervised) network connections associated with the outcomes and the multiple scores corresponding to their subnetwork estimates facilitate data interpretation. We estimated the functional networks between brain regions of each individual and applied discriminant analysis methods such as machine learning as a biomarker to assist diagnosis. The proposed score showed good disease prediction accuracy in both numerical experiments and real data analysis, and reasonable results were produced by the real data analysis.

In this study, a SSHCA method for constructing prognostic risk scores was proposed. The method could be run on the latest version of the R package msma and it was characterized by supervised learning to improve the prediction accuracy for scoring of the estimated network. The brain time series images had a spatio-temporal structure per person, and the spatial structure was transformed into a network structure, and the scoring process had a hierarchical structure. At the lower level, brain networks are estimated for each individual, and at the upper level, they are integrated to enable group analysis.

Moreover, a method to break up the hierarchical structure and make the network estimation independent was considered. A score was created after the network was given. Because the glasso method was useful and faster for network estimation, we incorporated the glasso into our algorithm to estimate the network. The method of calculating the score while estimating the network tended to be more accurate in predicting the score than the method of calculating the score after the network estimation was completed. Because their prediction accuracies were not very different, we considered that the independent estimation method offered a more precise network estimation. Furthermore, as the score is decomposed into multiple components like principal component analysis, data-driven network decomposition is possible and networks useful for diagnosis can be selected. Thus, the score

structure of the proposed method may allow for more detailed interpretation of the analysis results.

Despite these advantages, setting tuning parameters remains a challenge. Theoretically, it is possible to set many parameters to adjust sparseness, but this must be restricted if the sample size for training is small. In practice, we applied a simplification but it may be an open question as to how many parameters to set. Moreover, the same may be said for the tuning parameters used to adjust the degree of supervision. In the real data used in this study, the difference in SSHCA scores between the disease group and the healthy group was small and difficult to discriminate. This may be the reason why the tuning parameter $\mu = 1$ was chosen in the proposed method. It will be a challenge to investigate this in various stages of AD progression.

Although there are many machine learning methods, the focus of this study was to determine if the proposed network scores were useful as features. Although limited methods were applied for this reason, the simulation study and the analysis of the real data showed that all the methods produced scores that could be predicted with a certain degree of accuracy. In view of this, it was possible to estimate disease prediction probabilities with good disease prediction accuracy using simple methods, such as a multiple logistic regression model (with variable selection) when using the proposed scores. Such a simple model has been used in many clinical studies because they make interpretation of results easier and may be very useful for interpreting results without discussing explanatory possibilities in complex models.

Neural networks have been developed in recent years, and their deep learning has become increasingly useful in neuroimaging [16]. The graph neural network is specialized to perform the network analysis targeted in this study. This method takes a given network as input and requires the network to be estimated a priori. In the latest research [17], Pearson's correlation coefficient is used to estimate the network first, and then graph neural networks are applied. We attempted to improve upon the sequential approach used in existing methods, in which a regression analysis is performed after network estimation. Such a network estimation method based on correlation coefficients describes relationships between brain regions, and these relationships do not necessarily include the relationship with diseases. This is unlikely to directly improve the accuracy of the diagnostic or predictive models targeted in this study. It is expected that the accuracy will be improved if not only the network estimation but also the relationship with the disease is included in the scoring. The results of numerical experiments and applications in this paper show that scoring has a certain accuracy in diagnosis and network estimation simultaneously, and moreover, there was not much difference in terms of prediction accuracy between neural networks or other machine learning methods and simple logistic regression analysis. In general, neural networks are more likely to have complexity and still need stability in terms of interpretability. On the contrary, the scores of the proposed method have a linear structure, and the application of a linear logistic regression model to them ensures prediction accuracy, which is advantageous in that it preserves the interpretability of linearity. Nevertheless, the application of the proposed scores to graph neural networks is interesting and could be a future challenge.

The proposed method has many potential extensions, and it could be used to estimate directed networks with arrows between nodes. It could also be used for dynamic modeling, as mentioned in the methods section. To interpret the results, we needed to simplify the method and have discussions with experts, which is beyond the scope of this study. The scoring used in this method could be incorporated into the multiblock method by [29]. This method enabled us to evaluate the relationship between rs-

fMRI and other brain images, such as structural MRI, while scoring. This is called multimodal analysis and is one of the most important analyses in brain image analysis [31]. This has not been developed within the framework of the aforementioned graph neural networks, which is another useful aspect of our method. Scoring could be considered as a dimension reduction and could contribute not only to discrimination, but also to subtype classification by clustering. [32] performed network clustering on genetic data and [33] analyzed the relationship between structural MRI and estimated networks from non-imaging data such as CSF and blood biomarkers. Thus, this is expected to be a method of analysis that can be used to develop many brain studies.

Our method is applicable to other medical data as well, and may be useful for general clinical data, education and social medicine. Since our method has an element of dimension reduction, it is useful in high-dimensional data analysis, and a representative area is genetic data, which was also the subject of analysis in [32]. In recent years, with the development of measuring instruments, especially single-cell RNA-sequencing (scRNA-seq) measurements, it has become possible to make more precise measurements, identify known novel cell types and characterize gene-gene interactions within each cell type. Because of the heterogeneity revealed in scRNA-seq data among various cell types in the same tissue, cell-type-level gene networks are expected to reveal gene-gene interactions that have not been revealed in previous tissue-level gene networks. As described in [34], the method of analysis is closely related to the functional brain networks targeted in this study. Many statistically challenging issues have been pointed out, and it may be possible to develop the framework of our method toward these issues.

5. Conclusions

We developed a method to estimate diagnostic probability from rs-fMRI data using supervised and data-driven (sub) network estimation. The scoring method and its algorithm were introduced and simulation analysis and application to real brain imaging data revealed that the regression model with the created scores could predict diseases with higher accuracy. There are several potential extensions of this method, and future work is to apply it to various diseases and obtain new medical knowledge. Our method can assist in the construction of brain disease biomarkers from functional imaging data.

Acknowledgments

The author is deeply grateful to the referees for constructive comments. This study was supported in part by Intramural Research Grant (27-8) for Neurological and Psychiatric Disorders of NCNP, a Grant-in-Aid from the Ministry of Education, Culture, Sport, Science and Technology of Japan (24700286), and JST CREST (JPMJCR21D3). For this research, we used the supercomputer of ACCMS, Kyoto University.

Data collection and sharing for this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01 AG024904) and DOD ADNI (Department of Defense award number W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging, the National Institute of Biomedical Imaging and Bioengineering, and through generous contributions from the following: Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Araclon Biotech; BioClinica, Inc.; Biogen Idec Inc.; Bristol-Myers Squibb Company; Eisai Inc.; Elan

Pharmaceuticals, Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd and its affiliated company Genentech, Inc.; Fujirebio; GE Healthcare ; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development, LLC.; Johnson & Johnson Pharmaceutical Research & Development LLC.; Medpace, Inc.; Merck & Co., Inc.; Meso Scale Diagnostics, LLC.; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Synarc Inc.; and Takeda Pharmaceutical Company. The Canadian Institutes of Health Research provide funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (`www.fnih.org`). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Disease Cooperative Study at the University of California, San Diego. ADNI data are disseminated by the Laboratory for Neuro Imaging at the University of Southern California.

Data used in preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in the data analysis or writing of this report. A complete listing of ADNI investigators can be found at:
http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf

Conflict of interests

The authors declare that they have no financial or nonfinancial conflicts of interest.

References

1. S. Amemiya, H. Takao, O. Abe, Resting-State fMRI: Emerging Concepts for Future Clinical Application, *J. Magn. Reson. Imaging*, **2023** (2023). https://doi.org/10.1002/jmri.28894
2. S. H. Joo, H. K. Lim, C. U. Lee, Three large-scale functional brain networks from resting-state functional MRI in subjects with different levels of cognitive impairment, *Psychiatry Invest.*, **13** (2016).
3. A. Chase, Altered functional connectivity in preclinical dementia, *Nat. Rev. Neurol.*, **10** (2014), 11. https://doi.org/10.1038/nrneurol.2014.195
4. X. Chen, H. Zhang, Y. Gao, C. Y. Wee, G. Li, D. Shen, et al., High-order resting-state functional connectivity network for MCI classification, *Human Brain Mapp.*, **37** (2016), 3282–3296. https://doi.org/10.1002/hbm.23240
5. B. Ibrahim, S. Suppiah, N. Ibrahim, M. Mohamad, H. Hassan, N. Nasser, et al., Diagnostic power of resting-state fMRI for detection of network connectivity in Alzheimer's disease and mild cognitive impairment: A systematic review, *Human Brain Mapp.*, **42** (2021), 2941–2968. https://doi.org/10.1002/hbm.25369
6. S. L. Warren, A. A. Moustafa, Functional magnetic resonance imaging, deep learning, and Alzheimer's disease: A systematic review, *J. Neuroimaging*, **33** (2023), 5–18. https://doi.org/10.1111/jon.13063

7. S. Khan, A. Gramfort, N. R. Shetty, M. G. Kitzbichler, S. Ganesan, J. M. Moran, et al., Local and long-range functional connectivity is reduced in concert in autism spectrum disorders, *Proceed. Natl Acad. Sci.*, **110** (2013), 3107–3112. https://doi.org/10.1073/pnas.1214533110
8. H. Karbasforoushan, N. Woodward, Resting-state networks in schizophrenia, *Curr. Top. Med. Chem.*, **12** (2012), 2404–2414. https://doi.org/10.2174/156802612805289863
9. J. W. Murrough, C. G. Abdallah, A. Anticevic, K. A. Collins, P. Geha, L. A. Averill, et al., Reduced global functional connectivity of the medial prefrontal cortex in major depressive disorder, *Human Brain Mapp.*, **37** (2016), 3214–3223.
10. Z. Li, R. Chen, M. Guan, E. Wang, T. Qian, C. Zhao, et al., Disrupted brain network topology in chronic insomnia disorder: a resting-state fMRI study, *NeuroImage Clin.*, **18** (2018), 178–185. https://doi.org/10.1016/j.nicl.2018.01.012
11. M. Brown, G. S. Sidhu, R. Greiner, N. Asgarian, M. Bastani, P. H. Silverstone, et al., ADHD-200 Global Competition: diagnosing ADHD using personal characteristic data can outperform resting state fMRI measurements, *Front. Syst. Neurosci.*, **6** (2012), 69. https://doi.org/10.3389/fnsys.2012.00069
12. M. D. Rosenberg, E. S. Finn, D. Scheinost, X. Papademetris, X. Shen, R. T. Constable, et al., A neuromarker of sustained attention from whole-brain functional connectivity, *Nat. Neurosci.*, **19** (2016), 165–171. https://doi.org/10.1038/nn.4179
13. M. Rosenberg, E. Finn, D. Scheinost, R. Constable, M. Chun, Characterizing attention with predictive network models, *Trends Cognit. Sci.*, **21** (2017), 290–302. https://doi.org/10.1016/j.tics.2017.01.011
14. E. Finn, X. Shen, D. Scheinost, M. Rosenberg, J. Huang, M. Chun, et al., Functional connectome fingerprinting: identifying individuals using patterns of brain connectivity, *Nat. Neurosci.*, **18** (2015), 1664–1671. https://doi.org/10.1038/nn.4135
15. X. Shen, E. S. Finn, D. Scheinost, M. D. Rosenberg, M. M. Chun, X. Papademetris, et al., Using connectome-based predictive modeling to predict individual behavior from brain connectivity, *Nat. Protoc.*, **12** (2017), 506–518. https://doi.org/10.1038/nprot.2016.178
16. P. H. Luckett, J. J. Lee, K. Y. Park, R. V. Raut, K. L. Meeker, E. M. Gordon, et al., Resting state network mapping in individuals using deep learning, *Front. Neurol.*, **13** (2023), 1055437. https://doi.org/10.3389/fneur.2022.1055437
17. J. Gao, J. Liu, Y. Xu, D. Peng, Z. Wang, Brain age prediction using graph neural network based on resting-state functional MRI in Alzheimer's disease, *Front. Neurosci.*, **17** (Year), 1222751.
18. H. I. Suk, C. Y. Wee, S. W. Lee, D. Shen, State-space model with deep learning for functional dynamics estimation in resting-state fMRI, *NeuroImage*, **129** (2016), 292–307. https://doi.org/10.1016/j.neuroimage.2016.01.005
19. H. Du, M. Xia, K. Zhao, X. Liao, H. Yang, Y. Wang, et al., PAGANI Toolkit: Parallel graph-theoretical analysis package for brain network big data, *Human Brain Mapp.*, **39** (2018), 1869. https://doi.org/10.1002/hbm.23996

20. J. Liu, Y. Pan, F. X. Wu, J. Wang, Enhancing the feature representation of multi-modal MRI data by combining multi-view information for MCI classification, *Neurocomputing*, **400** (2020), 322–332. https://doi.org/10.1016/j.neucom.2020.03.006

21. C. Y. Wee, P. T. Yap, K. Denny, J. N. Browndyke, G. G. Potter, K. A. Welsh-Bohmer, et al., Resting-state multi-spectrum functional connectivity networks for identification of MCI patients, *PloS One*, **7** (2012), e37828. https://doi.org/10.1371/journal.pone.0037828

22. B. Jie, D. Zhang, W. Gao, Q. Wang, C. Wee, D. Shen, Integration of network topological and connectivity properties for neuroimaging classification, *IEEE Trans. Biomed. Eng.*, **61** (2013), 576–589. https://doi.org/10.1109/TBME.2013.2284195

23. X. Liang, J. Wang, C. Yan, N. Shu, K. Xu, G. Gong, et al., Effects of different correlation metrics and preprocessing factors on small-world brain functional networks: a resting-state functional MRI study, *PloS One*, **7** (2012), e32766. https://doi.org/10.1371/journal.pone.0032766

24. Y. Wang, J. Kang, P. B. Kemmer, Y. Guo, An efficient and reliable statistical method for estimating functional connectivity in large scale brain networks using partial correlation, *Front. Neurosci.*, **10** (2016), 123. https://doi.org/10.3389/fnins.2016.00123

25. Y. Li, Y. Wang, G. Wu, F. Shi, L. Zhou, W. Lin, et al., Discriminant analysis of longitudinal cortical thickness changes in Alzheimer's disease using dynamic and network features, *Neurobiol. Aging*, **33** (2012), e15–e30. https://doi.org/10.1016/j.neurobiolaging.2010.11.008

26. C. Y. Wee, P. T. Yap, D. Zhang, L. Wang, D. Shen, Group-constrained sparse fMRI connectivity modeling for mild cognitive impairment identification, *Brain Struct. Funct.*, **219** (2014), 641–656. https://doi.org/10.1007/s00429-013-0524-8

27. M. J. Rosa, L. Portugal, T. Hahn, A. J. Fallgatter, M. I. Garrido, J. Shawe-Taylor, et al., Sparse network-based models for patient classification using fMRI, *Neuroimage*, **105** (2015), 493–506. https://doi.org/10.1016/j.neuroimage.2014.11.021

28. J. Friedman, T. Hastie, R. Tibshirani, Sparse inverse covariance estimation with the graphical lasso, *Biostatistics*, **9** (2008), 432–441. https://doi.org/10.1093/biostatistics/kxm045

29. A. Kawaguchi, F. Yamashita, Supervised multiblock sparse multivariable analysis with application to multimodal brain imaging genetics, *Biostatistics*, **18** (2017), 651–665. https://doi.org/10.1093/biostatistics/kxx011

30. A. Kawaguchi, *Multivariate Analysis for Neuroimaging Data*, CRC Press, (2021). https://doi.org/10.1201/9780429289606

31. N. Chaari, H. C. Akdağ, I. Rekik, Comparative survey of multigraph integration methods for holistic brain connectivity mapping, *Med. Image Anal.*, **2023** (2023), 102741. https://doi.org/10.1016/j.media.2023.102741

32. A. Kawaguchi, R. Yamanaka, Gene expression signature-based prognostic risk score with network structure, *Primary Central Nerv. Syst. Lymphoma*, **2016** (2016), 67–80.

33. H. Yoshida, A. Kawaguchi, F. Yamashita, K. Tsuruya, The utility of a network–based clustering method for dimension reduction of imaging and non-imaging biomarkers predictive of Alzheimer's disease, *Sci. Rep.*, **8** (2018), 1–10. https://doi.org/10.1038/s41598-018-21118-1

34. Y. Wang, L. Li, J. J. Li, H. Huang, Network Modeling in Biology: Statistical Methods for Gene and Brain Networks, *Stat. Sci.*, **36** (2021),

35. H. Shen, J. Z. Huang, Sparse principal component analysis via regularized low rank matrix approximation, *J. Multivar. Anal.*, **99** (2008), 1015–1034. https://doi.org/10.1016/j.jmva.2007.06.007

Appendix

A diagram representing the analysis procedure in this study is shown in Figure A1. Particularly original is the scoring method and the algorithm used to derive it.

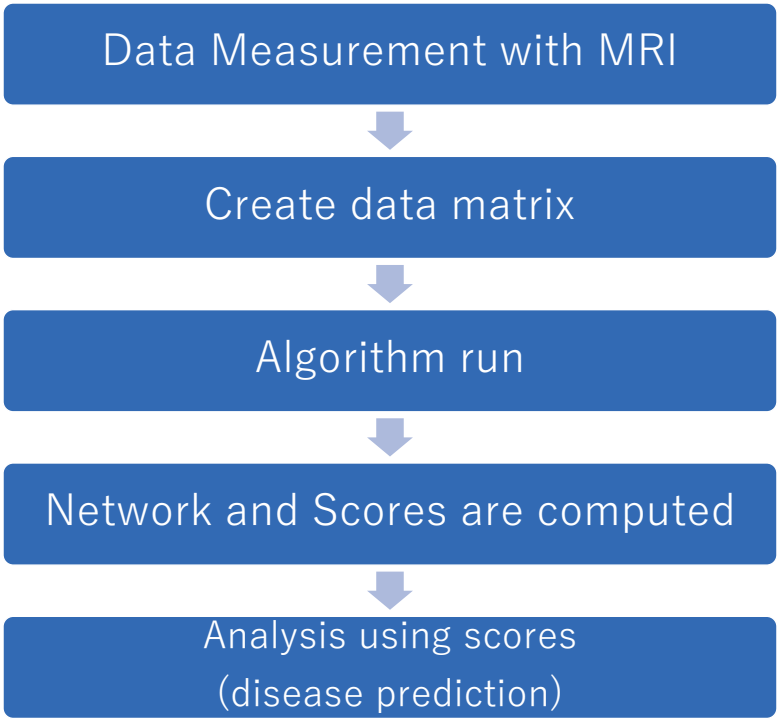

Figure A1. Diagram for the analysis procedure.

This section also provides the rationality of the algorithm in Section 2. First of all, we present a lemma provided by [35]. Let $\hat{\beta}$ be the minimizer of $\beta^2 - 2y\beta + p_\lambda(|\beta|)$. For the penalty $p_\lambda(|\theta|) = 2\lambda|\theta|$, the $\hat{\beta}$ is given by $\hat{\beta} = h_\lambda(y) = sign(y)(|y| - \lambda)_+$, where $(x)_+ = \max(0, x)$. This fact is used to derive a reasonable algorithm for solving the optimization problem.

Next, the notations are given as follows. Individual data is denoted by $T \times M$ matrix $\boldsymbol{X}_i = (\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \ldots, \boldsymbol{x}_{i,M})$, $i = 1, 2, \ldots, n$, $\boldsymbol{x}_{i,m} = (x_{i,m}(1), x_{i,m}(2), \ldots, x_{i,m}(T))^\top$. Sub data subtracting m-th column is denoted by $T \times (M-1)$ matrix $\boldsymbol{X}_{i,(-m)} = (\boldsymbol{x}_{i,1}, \ldots, \boldsymbol{x}_{i,m-1}, \boldsymbol{x}_{i,m+1}, \ldots, \boldsymbol{x}_{i,M})$. This can be represented with $\boldsymbol{X}_{i,(-m)} = (\boldsymbol{x}_{i,(-m)}(1), \boldsymbol{x}_{i,(-m)}(2), \ldots, \boldsymbol{x}_{i,(-m)}(T))^\top$ and $\boldsymbol{x}_{i,(-m)}(t) = (x_{i,1}(t), \ldots, x_{i,m-1}(t), x_{i,m+1}(t), \ldots, x_{i,M}(t))^\top$.

The scores have a hierarchical structure, and there are three types of scores: population, individual top and individual bottom. The population score is given by $\boldsymbol{s} = \boldsymbol{S}_2\boldsymbol{w}_2$ where $\boldsymbol{w}_2 = (w_{2,1}, w_{2,2}, \ldots, w_{2,M})^\top$, $\boldsymbol{S}_2$ is the $n \times M$ matrix with the (i, m)-element $s_{2,i,m}$ which is the individual

top score. $s_{2,i,m} = \sum_{t=1}^{T} s_{3,i,m}(t)w_{3,i}(t) = \boldsymbol{s}_{3,i,m}\boldsymbol{w}_{3,i}$ with $\boldsymbol{w}_{3,i} = (w_{3,i}(1), w_{3,i}(2), \ldots, w_{3,i}(T))^\top$ and the individual bottom score $s_{3,i,m}(t)$ defined by $s_{3,i,m}(t) = \boldsymbol{x}_{i,(-m)}^\top(t)\boldsymbol{w}_{4,i,m}$ and $\boldsymbol{w}_{4,i,m}$ is the n-dimensional weight vector. The hierarchical structure can be seen by writing down the scores as follows.

$$\boldsymbol{s} = \boldsymbol{S}_2\boldsymbol{w}_2 = \sum_{m=1}^{M} w_{2,m}\left\{\sum_{t=1}^{T} \boldsymbol{w}_3(t) \circ \boldsymbol{s}_{3,m}(t)\right\}$$

where $\boldsymbol{w}_3(t) = (w_{3,1}(t), w_{3,2}(t), \ldots, w_{3,n}(t))^\top$, $\boldsymbol{s}_{3,m}(t) = (s_{3,1,m}(t), s_{3,2,m}(t), \ldots, s_{3,n,m}(t))^\top$ and $\circ$ denotes the Hadamard product (the element-wise product). Using this notation, the individual bottom score can be rewritten as follows.

$$\boldsymbol{s}_{3,m}(t) = \boldsymbol{X}_{(-m)}(t)\boldsymbol{w}_{4,m}$$

where $\boldsymbol{w}_{4,m} = (\boldsymbol{w}_{4,1,m}^\top, \ldots, \boldsymbol{w}_{4,n,m}^\top)^\top$, $\boldsymbol{X}_{(-m)}(t) = \mathrm{diag}(\boldsymbol{x}_{1,(-m)}^\top(t), \boldsymbol{x}_{2,(-m)}^\top(t), \ldots, \boldsymbol{x}_{n,(-m)}^\top(t))$ is the $n \times n(M-1)$ matrix.

A.1. Optimization function

First, consider Eq 2.1 as the optimization problem $\max L_0(\boldsymbol{w})$ subject to $\|\boldsymbol{w}_2\|^2 = 1, \|\boldsymbol{w}_{3,i}\|^2 = 1$, $\|\boldsymbol{w}_{4,i,m}\|^2 = 1$ where

$$L_0(\boldsymbol{w}) = \{L_{01}(\boldsymbol{w}_2, \boldsymbol{w}_3, \boldsymbol{w}_4) - P_{2,\lambda_2}(\boldsymbol{w}_2)\} + \{L_{02}(\boldsymbol{w}_3, \boldsymbol{w}_4) - P_{3,\lambda_3}(\boldsymbol{w}_3)\} + \{L_{03}(\boldsymbol{w}_4) - P_{4,\lambda_4}(\boldsymbol{w}_4)\}$$

The first term on the right side of $L_0(\boldsymbol{w})$ is given by

$$L_{01}(\boldsymbol{w}_2, \boldsymbol{w}_3, \boldsymbol{w}_4) = (1-\mu) \times cov(\boldsymbol{s}, \boldsymbol{u}) + \mu \times cov(\boldsymbol{s}, \boldsymbol{Z})$$

with $0 \leq \mu \leq 1$. Note that we actually considered the covariance of $\boldsymbol{s}$ (i.e., the variance of $\boldsymbol{s}$), but the algorithm used an iterative calculation similar to the principal component analysis, where one score was fixed and the weight for the other score was calculated, then reversed and the weight was calculated again. For this purpose, we prepared a vector $\boldsymbol{u}$ which has the same length as $\boldsymbol{s}$. The covariance is defined as follows:

$$\begin{aligned} cov(\boldsymbol{s}, \boldsymbol{u}) = \boldsymbol{s}^\top\boldsymbol{u} &= \left(\sum_{m=1}^{M} w_{2,m}\boldsymbol{s}_{2,m}\right)^\top \boldsymbol{u} = \sum_{m=1}^{M} w_{2,m}\boldsymbol{s}_{2,m}^\top\boldsymbol{u} \\ &= \sum_{m=1}^{M} w_{2,m}\left\{\sum_{t=1}^{T} w_3^\top(t) \circ s_{3,m}^\top(t)\right\}\boldsymbol{u} \\ &= \sum_{m=1}^{M} w_{2,m}\left[\sum_{t=1}^{T} w_3^\top(t) \circ \{w_{4,m}^\top X_{(-m)}^\top(t)\}\right]\boldsymbol{u} \end{aligned}$$

Thus, we see that $L_{01}(\boldsymbol{w}_2, \boldsymbol{w}_3, \boldsymbol{w}_4)$ depends on $\boldsymbol{w}_2, \boldsymbol{w}_3$ and $\boldsymbol{w}_4$. The function $L_{02}(\boldsymbol{w}_3, \boldsymbol{w}_4)$ in the second term on $L_0(\boldsymbol{w})$ is given by

$$L_{02}(\boldsymbol{w}_3, \boldsymbol{w}_4) = \sum_{i=1}^{n} cov(\boldsymbol{s}_{2,i}, \boldsymbol{u}_{2,i})$$

where $\boldsymbol{s}_{2,i}$ is the M-dimensional vector with the m-th element $s_{2,i,m}$. As in the first term, prepared a vector $\boldsymbol{u}_{2,i}$ of the same length as $\boldsymbol{s}_{2,i}$ for the iterative calculation. The covariance is given by $cov(\boldsymbol{s}_{2,i}, \boldsymbol{u}_{2,i}) = \boldsymbol{s}_{2,i}^\top \boldsymbol{u}_{2,i} = \boldsymbol{w}_{3,i}^\top \boldsymbol{S}_{3,i}^\top \boldsymbol{u}_{2,i}$ where $\boldsymbol{S}_{3,i}$ is the $T \times M$ matrix with the (t, m)-element $s_{3,i,m}(t) = \boldsymbol{x}_{i,(-m)}^\top(t)\boldsymbol{w}_{4,i,m}$. Thus, we see that $L_{02}(\boldsymbol{w}_3, \boldsymbol{w}_4)$ depends on $\boldsymbol{w}_3$ and $\boldsymbol{w}_4$.

The function $L_{03}(\boldsymbol{w}_4)$ in the third term on $L_0(w)$ is given by

$$L_{03}(\boldsymbol{w}_4) = \sum_{i=1}^{n}\sum_{m=1}^{M} cov(\boldsymbol{s}_{3,i,m}, \boldsymbol{x}_{im})$$

where $\boldsymbol{s}_{3,i,m}$ is the T-dimensional vector with the t-th element $s_{3,i,m}(t) = \boldsymbol{x}_{i,(-m)}^\top(t)\boldsymbol{w}_{4,i,m}$, and the is covariance defined as $cov(\boldsymbol{s}_{3,i,m}, \boldsymbol{x}_{im}) = \boldsymbol{s}_{3,i,m}^\top \boldsymbol{x}_{im} = \boldsymbol{w}_{4,i,m}^\top \boldsymbol{X}_{i,(-m)}^\top \boldsymbol{x}_{im}$. Thus, we see that $L_{03}(\boldsymbol{w}_4)$ depends on only $\boldsymbol{w}_4$.

The weights of each term were regularized.

$$P_{2,\lambda_2}(\boldsymbol{w}_2) = P_{\lambda_2}(\boldsymbol{w}_2), \quad P_{3,\lambda_3}(\boldsymbol{w}_3) = \sum_{i=1}^{n} P_{\lambda_{3,i}}(w_{3,i}), \quad P_{4,\lambda_4}(\boldsymbol{w}_4) = \sum_{i=1}^{n}\sum_{m=1}^{M} P_{\lambda_{4,i,m}}(w_{4,i,m})$$

$P_\lambda(x) = 2\lambda|x|$ and $P_\lambda(\boldsymbol{x}) = 2\lambda \sum_j |x_j| = \sum_j P_\lambda(x_j)$.

A.1.1. Optimization

In order to solve the optimization problem max $L_0(w)$ subject to $\|\boldsymbol{w}_2\|^2 = 1, \|\boldsymbol{w}_{3,i}\|^2 = 1$, $\|\boldsymbol{w}_{4,i,m}\|^2 = 1$, we consider the Lagrangian optimization problem max $L_1(\boldsymbol{w})$, where

$$L_1(\boldsymbol{w}) = L_0(\boldsymbol{w}) - \eta_2\|\boldsymbol{w}_2\|^2 - \sum_{i=1}^{n}\eta_{3,i}\|\boldsymbol{w}_{3,i}\|^2 - \sum_{i=1}^{n}\sum_{m=1}^{M}\eta_{4,i,m}\|\boldsymbol{w}_{4,i,m}\|^2$$

with $\eta_2 > 0$, $\eta_{3,i} > 0$ and $\eta_{4,i,m} > 0$ are the Lagrange multiplier.

A population weight $\boldsymbol{w}_2$ for $\boldsymbol{X}$ is obtained by considering the following objective function:

$$\begin{aligned}
L_{11}(\boldsymbol{w}_2) &= L_{01}(\boldsymbol{w}_2, \boldsymbol{w}_3, \boldsymbol{w}_4) - P_{\lambda_2}(\boldsymbol{w}_2) - \eta_2\|\boldsymbol{w}_2\|^2 \\
&= (1-\mu) \times cov(\boldsymbol{s}, \boldsymbol{u}) + \mu \times cov(\boldsymbol{s}, \boldsymbol{Z}) - P_{\lambda_2}(\boldsymbol{w}_2) - \eta_2\|\boldsymbol{w}_2\|^2 \\
&= \sum_{m=1}^{M} w_{2,m}\boldsymbol{s}_{2,m}^\top\{(1-\mu) \times \boldsymbol{u} + \mu \times \boldsymbol{Z}\} - P_{\lambda_2}(\boldsymbol{w}_2) - \eta_2\|\boldsymbol{w}_2\|^2 \\
&= \sum_{m=1}^{M}\left[w_{2,m}\boldsymbol{s}_{2,m}^\top\{(1-\mu) \times \boldsymbol{u} + \mu \times \boldsymbol{Z}\} - P_{\lambda_2}(w_{2,m}) - \eta_2 w_{2,m}^2\right]
\end{aligned}$$

From the Lemma, $\boldsymbol{w}_2$ such that $L_{11}(\boldsymbol{w}_2)$ is maximized is as follows.

$$\text{argmax}L_{11}(\boldsymbol{w}_2) = \frac{1}{\eta_2} h_{\lambda_{2m}}(\boldsymbol{s}_{2,m}^\top((1-\mu) \times \boldsymbol{u} + \mu \times \boldsymbol{Z})) =: \tilde{\boldsymbol{w}}_2$$

The final estimate is obtained by normalizing $\hat{\boldsymbol{w}}_2 = \tilde{\boldsymbol{w}}_2/\|\tilde{\boldsymbol{w}}_2\|$.

Similarly, an individual top weight $\boldsymbol{w}_3$ for $\boldsymbol{X}$ is obtained by considering the following objective function:

$$\begin{aligned} L_{12}(\boldsymbol{w}_3) &= L_{01}(\boldsymbol{w}_2, \boldsymbol{w}_3, \boldsymbol{w}_4) + L_{02}(\boldsymbol{w}_3, \boldsymbol{w}_4) - P_{\lambda_3}(\boldsymbol{w}_3) - \sum_{i=1}^{n} \eta_{3,i} \|\boldsymbol{w}_{3,i}\|^2 \\ &= (1-\mu) \times cov(\boldsymbol{s}, \boldsymbol{u}) + \mu \times cov(\boldsymbol{s}, \boldsymbol{Z}) + \sum_{i=1}^{n} \{cov(\boldsymbol{s}_{2,i}, \boldsymbol{u}_{2,i}) - P_{\lambda_{3,i}}(\boldsymbol{w}_{3,i})\} - \sum_{i=1}^{n} \eta_{3,i} \|\boldsymbol{w}_{3,i}\|^2 \end{aligned}$$

The individual top weight $\boldsymbol{w}_3$ depends on functions $L_{01}(\boldsymbol{w}_2, \boldsymbol{w}_3, \boldsymbol{w}_4)$ and $L_{02}(\boldsymbol{w}_3, \boldsymbol{w}_4)$. We write this expression for each element i.

$$L_{121}(\boldsymbol{w}_{3,i}) := \sum_{m=1}^{M} \{w_{2,m}((1-\mu)u_i + \mu Z_i)\boldsymbol{s}_{3,i,m}\boldsymbol{w}_{3,i}\} + \boldsymbol{w}_{3,i}^{\top}\boldsymbol{S}_{3,i}^{\top}\boldsymbol{u}_{2,i} - P_{\lambda_{3,i}}(\boldsymbol{w}_{3,i}) - \eta_{3,i}\|\boldsymbol{w}_{3,i}\|^2$$

From the Lemma, $\boldsymbol{w}_{3,i}$ such that $L_{121}(\boldsymbol{w}_{3,i})$ is maximized is as follows.

$$\text{argmax} L_{121}(\boldsymbol{w}_{3,i}) = \frac{1}{2\eta_{3,i}} h_{\lambda_{2m}}\left(\sum_{m=1}^{M} \{w_{2,m}((1-\mu)u_i + \mu Z_i)\boldsymbol{s}_{3,i,m}\} + \boldsymbol{S}_{3,i}^{\top}\boldsymbol{u}_{2,i}\right) := \tilde{\boldsymbol{w}}_{3,i}$$

and the final estimate is obtained by normalizing $\hat{\boldsymbol{w}}_{3,i} = \tilde{\boldsymbol{w}}_{3,i}/\|\tilde{\boldsymbol{w}}_{3,i}\|$.

A individual bottom weight $\boldsymbol{w}_4$ for $\boldsymbol{X}$ is obtained by considering the following objective function:

$$\begin{aligned} L_{131}(\boldsymbol{w}_4) &= L_{01}(\boldsymbol{w}_2, \boldsymbol{w}_3, \boldsymbol{w}_4) + L_{02}(\boldsymbol{w}_3, \boldsymbol{w}_4) + L_{03}(\boldsymbol{w}_4) - P_{\lambda_4}(\boldsymbol{w}_4) - \sum_{i=1}^{n}\sum_{m=1}^{M} \eta_{4,i,m}\|\boldsymbol{w}_{4,i,m}\|^2 \\ &= (1-\mu)cov(\boldsymbol{s}, \boldsymbol{u}) + \mu \times cov(\boldsymbol{s}, \boldsymbol{Z}) + \sum_{i=1}^{n} cov(\boldsymbol{s}_{2,i}, \boldsymbol{u}_{2,i}) \\ &\quad + \sum_{i=1}^{n}\sum_{m=1}^{M} \{cov(\boldsymbol{s}_{3,i,m}, \boldsymbol{x}_{im}) - P_{\lambda_{4,i,m}}(\boldsymbol{w}_{4,i,m})\} - \sum_{i=1}^{n}\sum_{m=1}^{M} \eta_{4,i,m}\|\boldsymbol{w}_{4,i,m}\|^2 \end{aligned}$$

The individual bottom weight $\boldsymbol{w}_4$ depends on functions $L_{01}(\boldsymbol{w}_2, \boldsymbol{w}_3, \boldsymbol{w}_4)$, $L_{02}(\boldsymbol{w}_3, \boldsymbol{w}_4)$ and $L_{03}(\boldsymbol{w}_4)$.

There are several possible optimizations for $\boldsymbol{w}_4$. Since the first is the so-called the coordinate descent method, we refer to it as "corde" for short. For "corde", we rewrite $L_{13}(\boldsymbol{w}_4)$ by each elements i and m.

$$\begin{aligned} L_{1311}(\boldsymbol{w}_{4,i,m}) &:= w_{2,m}((1-\mu)\boldsymbol{u}_i + \mu\boldsymbol{Z}_i)\boldsymbol{w}_{3,i}^{\top}\boldsymbol{X}_{i,(-m)}\boldsymbol{w}_{4,i,m} + \boldsymbol{u}_{2,i,m}\boldsymbol{w}_{3,i}^{\top}\boldsymbol{X}_{i,(-m)}w_{4,i,m} \\ &\quad + \boldsymbol{x}_{im}^{\top}\boldsymbol{X}_{i,(-m)}\boldsymbol{w}_{4,i,m} - P_{\lambda_{4,i,m}}(\boldsymbol{w}_{4,i,m}) - \eta_{4,i,m}\|w_{4,i,m}\|^2 \\ &= [\{((1-\mu)u_i + \mu Z_i)w_{2,m} + u_{2,i,m}\}\boldsymbol{w}_{3,i}^{\top} + \boldsymbol{x}_{im}^{\top}]\boldsymbol{X}_{i,(-m)}\boldsymbol{w}_{4,i,m} - P_{\lambda_{4,i,m}}(\boldsymbol{w}_{4,i,m}) - \eta_{4,i,m}\|\boldsymbol{w}_{4,i,m}\|^2 \\ &= \sum_{m_2 \neq m} \{\{((1-\mu)u_i + \mu Z_i)w_{2,m} + u_{2,i,m}\}\boldsymbol{w}_{3,i}^{\top} + \boldsymbol{x}_{im}^{\top}\}\boldsymbol{x}_{i,m_2}w_{4,i,m,m_2} \\ &\quad - P_{\lambda_{4,i,m}}(w_{4,i,m,m_2}) - \eta_{4,i,m}w_{4,i,m,m_2}^2\} \end{aligned}$$

From the Lemma, $\boldsymbol{w}_{4,i,m}$ such that $L_{1311}(\boldsymbol{w}_{4,i,m})$ is maximized is as follows.

$$\text{argmax} L_{1311}(\boldsymbol{w}_{4,i,m}) = \frac{1}{2\eta_{4,i,m}} h_{\lambda_{4,i,m}}\left(\boldsymbol{X}_{i,(-m)}^{\top}[\{((1-\mu)u_i + \mu Z_i)w_{2,m} + u_{2,i,m}\}\boldsymbol{w}_{3,i} + \boldsymbol{x}_i]\right) =: \tilde{\boldsymbol{w}}_{4,i,m}$$

Thus, the final estimate is obtained by normalizing $\hat{\boldsymbol{w}}_{4,i,m} = \tilde{\boldsymbol{w}}_{4,i,m}/\|\tilde{\boldsymbol{w}}_{4,i,m}\|$.

In the expression $L_{01}(\boldsymbol{w}_2, \boldsymbol{w}_3, \boldsymbol{w}_4) + L_{02}(\boldsymbol{w}_3, \boldsymbol{w}_4) + L_{03}(\boldsymbol{w}_4)$ of the objective function $L_{13}(\boldsymbol{w}_4)$ for $\boldsymbol{w}_4$, the upper level weights $\boldsymbol{w}_2$ and $\boldsymbol{w}_3$ are dependent. Ignoring this dependency, we consider an objective function with only $\boldsymbol{w}_4$, then the alternative object function for the bottom weight $\boldsymbol{w}_4$ is given as follows.

$$\begin{aligned} L_{132}(\boldsymbol{w}_4) &= L_{03}(\boldsymbol{w}_4) - P_{\lambda_4}(\boldsymbol{w}_4) - \sum_{i=1}^{n}\sum_{m=1}^{M}\eta_{4,i,m}\|\boldsymbol{w}_{4,i,m}\|^2 \\ &= \sum_{i=1}^{n}\sum_{m=1}^{M}\{cov(\boldsymbol{s}_{3,i,m}, \boldsymbol{x}_{im}) - P_{\lambda_{4,i,m}}(\boldsymbol{w}_{4,i,m})\} - \sum_{i=1}^{n}\sum_{m=1}^{M}\eta_{4,i,m}\|\boldsymbol{w}_{4,i,m}\|^2 \end{aligned}$$

The corresponding "corde" in this objective function is referred to as "corde.ind", and "glasso" can also be applied. These details are given below.

For "corde.ind", we rewrite $L_{13}(\boldsymbol{w}_4)$ by each elements i and m.

$$\begin{aligned} L_{1321}(\boldsymbol{w}_{4,i,m}) &:= \boldsymbol{x}_{im}^{\top}\boldsymbol{X}_{i,(-m)}\boldsymbol{w}_{4,i,m} - P_{\lambda_{4,i,m}}(\boldsymbol{w}_{4,i,m}) - \eta_{4,i,m}\|\boldsymbol{w}_{4,i,m}\|^2 \\ &= \sum_{m_2 \neq m}\{x_{im}^{\top}x_{i,m_2}w_{4,i,m,m_2} - P_{\lambda_{4,i,m}}(w_{4,i,m,m_2}) - \eta_{4,i,m}w_{4,i,m,m_2}^2\} \end{aligned}$$

From the Lemma, $\boldsymbol{w}_{4,i,m}$ such that $L_{1321}(\boldsymbol{w}_{4,i,m})$ is maximized is as follows.

$$\text{argmax}L_{1321}(\boldsymbol{w}_{4,i,m}) = \frac{1}{2\eta_{4,i,m}}h_{\lambda_{4,i,m}}(\boldsymbol{X}_{i,(-m)}^{\top}\boldsymbol{x}_i) =: \tilde{\boldsymbol{w}}_{4,i,m}$$

Thus, the final estimate is obtained by normalizing $\hat{\boldsymbol{w}}_{4,i,m} = \tilde{\boldsymbol{w}}_{4,i,m}/\|\tilde{\boldsymbol{w}}_{4,i,m}\|$.

The "corde" was based the both functions $L_{131}(w_4)$ and $L_{132}(w_4)$. If the network estimation was to be implemented independently, then a graphical lasso solution would be available for the network estimation. The "glasso" is based the only function $L_{132}(w_4)$. For the variance-covariance matrix Σ_i and the sample variance-covariance matrix $\hat{\Sigma}_i = \boldsymbol{X}_i^{\top}\boldsymbol{X}_i$, the subfunction $L_{1323}(w)$ of objective function $L_{132}(w_4)$ is given as follows.

$$L_{1322}(w_{4,i,m}) = \frac{1}{2}\|\boldsymbol{y}_{i,m} - \Sigma_i^{\frac{1}{2}}\boldsymbol{w}_{4,i,m}\|^2$$

where $\boldsymbol{y}_{i,m} = \Sigma_i^{-1/2}\hat{\sigma}_{i,m}$ and $\boldsymbol{w}_{4,i,m} = \Sigma_i^{-1}\sigma_{i,m}$, and $\sigma_{i,m}$ and $\hat{\sigma}_{i,m}$ are the mth column elements of Σ_i and $\hat{\Sigma}_i$, respectively. The repeated optimization is applied until Σ and Σ_i^{-1} converge. See [28] for more details. This method was referred as "SSHCA-glasso".

Table A1. Simulation study results for *nsample* = 100, *nedge* = 100 and *ncomp* = 4.

nedgedif	Methods	μ	pathauc	scorecvauc	allmean
5(5%)	SSHCA-corde	0.0	0.782	0.513	0.692
		0.5	0.823	0.879	0.842
		1.0	0.836	0.991	0.888
	SSHCA-corde.ind	0.0	0.871	0.528	0.757
		0.5		0.730	0.824
		1.0		0.983	0.909
	SSHCA-glasso	0.0	0.907	0.518	0.777
		0.5		0.699	0.837
		1.0		0.983	0.932
	glasso		0.720	0.541	0.660
20(20%)	SSHCA-corde	0.0	0.781	0.504	0.689
		0.5	0.808	0.874	0.830
		1.0	0.816	0.992	0.875
	SSHCA-corde.ind	0.0	0.852	0.553	0.753
		0.5		0.721	0.809
		1.0		0.983	0.896
	SSHCA-glasso	0.0	0.872	0.541	0.761
		0.5		0.756	0.833
		1.0		0.979	0.907
	glasso		0.707	0.625	0.680

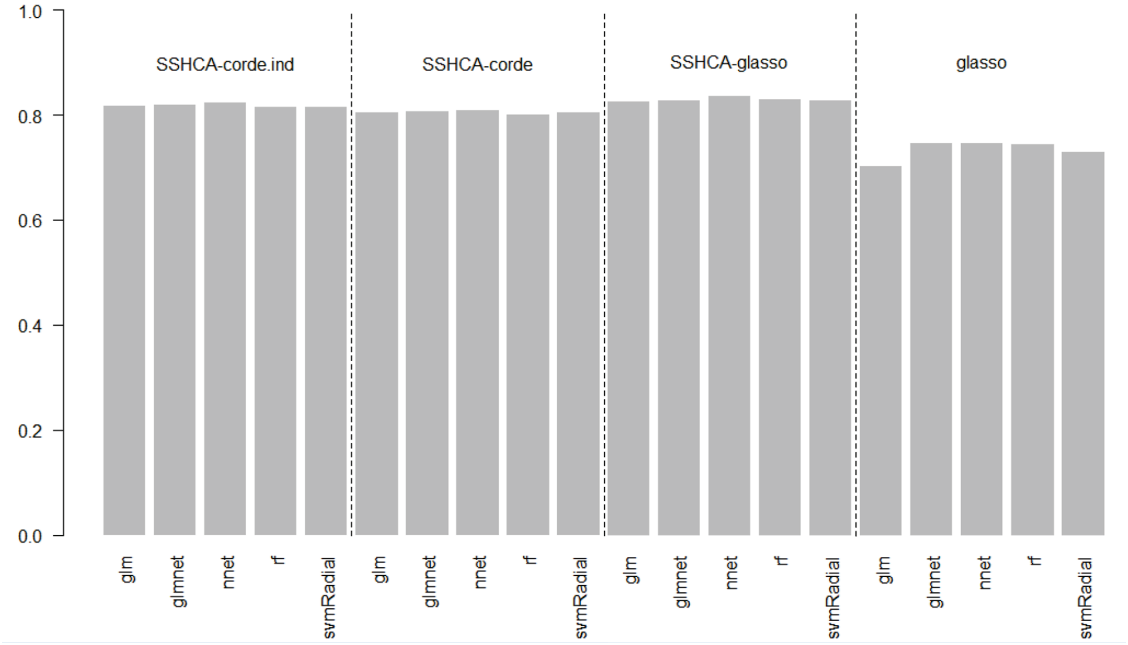

Figure A2. Results of simulation studies for different regression models.

A.1.2. Additional simulation results

Here, we showed the results when nsample was changed to 100 using the same settings as in the simulation study with results on the Tables 2 and 3. As nsample was increased, the overall value became better.

Other results of the simulation study are illustrated in Figure A2. The results for the regression models were averaged in the text, but here we have illustrated the results for each regression model when averaged over the other simulation parameters. The models were generalized linear model (glm), glmnet, support vector machine (svmRadial), random forests (rf) and neural networks (nnet). The results show that the prediction accuracy is not so different among the regression models.

Table A2. Simulation study results for nsample=100, nedge=50 and ncomp=4.

nedgedif	Methods	μ	pathauc	scorecvauc	allmean
5(10%)	SSHCA-corde	0.0	0.904	0.512	0.773
		0.5	0.907	0.792	0.869
		1.0	0.918	0.979	0.938
	SSHCA-corde.ind	0.0		0.522	0.804
		0.5	0.944	0.698	0.862
		1.0		0.986	0.958
	SSHCA-glasso	0.0		0.508	0.800
		0.5	0.946	0.659	0.850
		1.0		0.961	0.951
	glasso		0.926	0.542	0.798
20(40%)	SSHCA-corde	0.0	0.887	0.544	0.772
		0.5	0.890	0.846	0.875
		1.0	0.896	0.986	0.926
	SSHCA-corde.ind	0.0		0.525	0.761
		0.5	0.880	0.672	0.810
		1.0		0.965	0.908
	SSHCA-glasso	0.0		0.565	0.815
		0.5	0.940	0.757	0.879
		1.0		0.979	0.953
	glasso		0.909	0.686	0.835