Applied Intelligence
https://doi.org/10.1007/s10489-021-02877-3

A connected network-regularized logistic regression model for feature selection

Lingyu Li[1] · **Zhi-Ping Liu**[1]

Accepted: 27 September 2021

Abstract
Feature selection on a network structure can not only discover interesting variables but also mine out their intricate interactions. Regularization is often employed to ensure the sparsity and smoothness of the coefficients in logistic regression. However, currently available methods fail to embed the network connectivity in regularized penalty functions. In this paper, a connected network-regularized logistic regression (CNet-RLR) model for feature selection considering the structural connectivity in a network was proposed. Mathematically, it was a convex optimization problem constrained by inequalities reflecting network connectivity. Considering the non-differentiability of Lasso penalty, we constructed an equivalent formulation of CNet-RLR by employing auxiliary variables. An interior-point algorithm was designed to efficiently achieve the solutions. Theoretically, we proved their grouping effect and oracle property and guaranteed algorithmic convergence. In both synthetic simulation data and real-world uterine corpus endometrial carcinoma (UCEC) cancer genomics data, we validated the CNet-RLR model is efficient to identify the connected-network-structured features that can serve as diagnostic biomarkers. In the comparison study, we also proved the proposed CNet-RLR model results in better classification performance and feature interpretability than the other regularized logistic regression (RLR) alternatives and another graph embedded feature selection model.

Keywords Regularized logistic regression · Feature selection · Network-based sparse penalty · Network connectivity · Biomarker discovery

1 Introduction

Feature selection aims to identify critical variables in a specified feature set that are most relevant to a machine learning task [1]. At the preliminary stage, few domains explore more than one hundred features [2]. For different application scenarios, people have developed feature/variable subset selection strategies such as filter method, wrapper method and embedding method [3]. However, the situation has changed considerably with the creation of big data such as the emergence of high-throughput omics techniques [4]. For instance, microarray and RNA sequencing (RNA-seq) quantify the expression profiles of thousands of genes in parallel manners [5]. A wealth of new problems arise with the requirements for performing good learning and prediction, and novel ways to feature selection are in urgent demand [6, 7]. For instance in bioinformatics and machine learning, feature selection for the discovery of candidate biomarkers and disease genes plays important role in deciphering high-throughput data, avoiding dimensional disasters and improving prediction performance [8]. In precision medicine, it is rather a common request to select the most relevant feature genes from all the genes in human genome [9]. Moreover, genes certainly perform their functions by interacting with other genes in a form of pathway or a network [10].

Logistic regression (LR) is a supervised machine learning method with the dependent variables in discrete values [11]. Because of its easy interpretability [12], LR has played a huge role in problems related to binary classification viz. 0 and 1 [13, 14]. Nevertheless, the original LR model does not own feature selection ability such that it often causes a serious over-fitting problem in

✉ Zhi-Ping Liu
zpliu@sdu.edu.cn

Lingyu Li
lingyuli@mail.sdu.edu.cn

[1] School of Control Science and Engineering, Shandong University, Jinan 250061, China

Published online: 29 January 2022

prediction for multidimensional data [15]. Mathematically speaking, under the assumption that the number of samples is n and the number of variables is p, when $p >> n$, it becomes an ill-posed or ill-conditioned problem [16]. In the 1950s, Tikhonov proposed a method called regularization to address the ill-posed problem [17]. For improving its generalization ability while still maintaining the interpretability of LR, regularized logistic regression (RLR) models with various penalty functions, e.g., convex penalty, non-convex penalty and graph penalty, have been proposed and successively applied to feature selection and pattern recognition [18].

Convex regularization terms mainly include ridge [19], Lasso [20] and Elastic net (Enet for short) penalties [21]. Among them, ridge regression only shrinks the regression coefficients to ensure the relative smoothness between coefficients for solving their multicollinearity, that is, the regression coefficients will not be too large or too small, but will not be zeros [22]. Thus, the regularization term is not a good fit for feature reduction. Lasso term uses the L_1-norm to shrink the regression coefficients to zeros, then the variables with non-zero coefficients are selected, thereby generating a sparse solution. Enet penalty is a weighted form of the former two penalties, with smoothness and sparsity at the same time [21]. Currently, Enet is recognized as one of the cutting-edge interpretable feature selection methods [18].

Non-convex regularization terms typically include four functions: L_0 penalty [23], $L_{1/2}$ penalty [24], SCAD [25] and MCP [26]. Compared with convex-penalized methods, they tend to generate more sparse solutions for their direct constraints of variable numbers. However, their classification performances are not as perfect as Enet regression [27]. They cannot guarantee a global minimizer as promised by convex regression for any local minimizer. It is worth mentioning that $L_{1/2}$, SCAD and MCP regressions have oracle property [24, 28]. In contrast, ridge, Enet, and L_0 regularization methods do not have oracle property. Particularly, Lasso regression is proved to have asymptotic oracle property [29].

However, various RLR models with those convex or non-convex penalties mentioned above are limited to not using any previous knowledge or information regarding these features [30]. The available models were developed only from the perspective of calculation or algorithm [30, 31]. The features they singled out cannot express the information or significance they should have [32]. The RLR models don't think over the underlying interrelations between these selected variables when selecting features, but just calculate and shrink the unknown coefficients step by step according to a certain iterative algorithm for solving an optimization problem [30]. For example, the number of features determined by the Lasso regression method is related to the total number of variables contained in the experimental data [30]. Many kinds of literature have pointed out that it rarely selects features with strong correlation, but prefers to choose one representative feature of all ones with a strong correlation to construct a predictive model [21, 26, 33, 34]. Although such a model may have achieved good predictions, it only obtains some isolated features without any correlation. The interpretability of RLR will be definitely lost in case of selecting nonsense features.

High-throughput data such as genomics has become ubiquitous and extremely useful in analyzing various phenomena in biomedicine, but it is often difficult to establish a linkage between the analysis results (usually as a list of selected feature genes) and their functional significance [32]. Integrating a priori network-based knowledge is helpful in the high-throughput data analytics and the interpretation of discovering biomarkers or disease genes [26, 31]. Genes usually work collaboratively, as everyone knows, many cancer-related genes participate in an integrated pathway [35]. As mentioned, the prior knowledge of interacting network structure formulates a functional map of genes. So the feature selection for identifying biomarker genes needs to consider the structured pattern underlying these features and begin with these networked variables [36].

The regularization term of graph penalty function is such a kind of endeavor of introducing the networked structure in the feature selection [10, 34]. In mathematics, the essence of graph penalty function is to use the Laplacian matrix to represent the entire structure of a prior network/graph [37]. In physics, the Laplacian matrix of a graph is originally derived from that the gas in different areas diffuses from the high-density area to the low-density area driven by relative pressure [38]. However, in statistical learning, the importance of graph Laplacian regularization goes far beyond the diffusion phenomenon [39]. In terms of geometry, graph Laplacian matrix provides more information about the network structure [40]. In terms of algebra, its eigenvalues and eigenvectors are closely related to network connectivity [41].

Li and Li [30] proposed a network-constrained regularized logistic regression (Net-RLR) model to select features by analyzing genomics data. Although their work has caused controversy because Binder and Schumacher [42] argued that it does not take into account the variability in feature selection and lacks the comparison results with an empty model, that is, Net-RLR does not use any covariate information [39]. In our opinion, we need to pay much more attention to the research motivation. It aims to incorporate a priori network information into RLR model so as to obtain more biologically interpretable feature genes in the context of the knowledge about gene interactions, instead of getting more accurate prediction results [43]. In other words, we

can win the interpretability of these selected feature genes by sacrificing a little bit of prediction accuracy.

Since then, several network-based regularization methods have been proposed for modeling different feature networks or graphs. Mei et al. [44] proposed a network-regularized statistical topic model (NetSTM), which formally defines the topic modeling problem based on network structure. NetSTM uses the data-driven harmonic regularizer founded on the graph structure to regularize the statistical topic model, in which the graph harmonic function is induced by weights of certain a topic and graph Laplacian matrix [44]. In order to study the graph-regularized problem in multi-task learning, Zhou et al. [45] used a graph to take on the relationship between tasks, where each task is represented by a node and graph penalty is applied to express the difference between all nodes connected in the graph. Similar to Net-RLR model, Zhang et al. [31] applied network-regularized logistic model and protein-protein interaction (PPI) network to identify pathways regulated by biomolecules by mining gene expression profiles data, among which the graph penalty induced by the Laplacian matrix is defined as a generalized L_2-norm penalty. Recently, Wu et al. [46] also constructed a RLR method with network-based pairwise interaction to try to select robust biomarkers by integrating the degree information of PPI network into adaptive Enet. But, it only considered the adjacency or interaction between the paired (only two) genes. The selected genes result in discrete subnetworks composed of node pairs. Similarly, Min et al. [47] formulated several variants of network-regularized sparse LR model into an integrative model and gave the network-regularized LR model with absolute operation (AbNet-RLR) to estimate the coefficients with opposite signs.

The previous graph penalty terms include the prior information of the network in regularization, but they did not consider the connectivity of the network [33]. In fact, the features still have not selected sequentially guiding by a priori network structure. That is to say, the selection procedure does not work on the feature graph. The solutions often result in numerous disconnected components, even isolated features. As we all know, connected components are important structures of graphs or networks, and connectivity is an important property [15, 37, 38]. Connectivity represents integrative functional essentiality between nodes. In brain science image processing, considering the interactions between connected brain areas or regions would prove that there is an inherent relationship between the structure and function of the brain [33]. In forest planning, when selecting a wildlife habitat protection area, in order to maximize the satisfaction of species in mating, breeding, predation and so on, the most scientific approach is to choose a connected forest area as a protected area [48]. Similarly in genomics research, genes in high-throughput genomic data may contain some important interacting structures, the ideal gene selection strategy ought to consider the network-structured connectivity.

Ng et al. [33] proposed a connectivity-like sparse classifier and applied it to functional magnetic resonance imaging (fMRI) brain decoding. On world-real datasets, they proved that compared with standard classifiers, integrating connectivity information can not only improve the classification accuracy for diffusion tensor imaging data but also obtain a more interpretable weight pattern for fMRI. However, what we should point out here is that although they have claimed the fact of connectivity is crucial in the network structure, the proposed sparse classifier is essentially the graph penalty term proposed by Li and Li [30], which is just a different form of expression of mathematical symbols. In order to make the selected features have tight connectivity in the subnetwork, Kong and Yu [49] integrated the PPI network information into the deep neural network [50] using the adjacency matrix as the medium and presented a graph embedded deep feedforward networks (GEDFN) model. The experimental results on two RNA-seq datasets of tumor tissues showed that most of the selected genes form into a connected network. However, the entire network still contains many connected components. That is to say, the network connectivity constraints have not been fully embedded in the penalty function of regularization.

Moreover, Carvajal et al. [48] maintained large contiguous patches of mature forest in harvest scheduling models. In which, the node cut set of the graph are used to impose network connectivity constraints on the forest planning model so that the stands to be harvested or protected meet some form of structural connectivity. Similarly, Álvarez-Miranda and Sinnl [51] introduced a node-separator method (another version of node cut set) and used integer linear programming (ILP) to constrain the connectivity of networks. Althaus et al. [52] proposed an ILP method of detecting the maximum connected subgraph in a simple graph by introducing binary variables. Then, Li et al. [53] proposed a topology optimization model with connectivity called the virtual scalar field method (VSFM) and applied it to satisfy desired connectivity requirements in additive manufacturing or casting. Beyond that, they also provided a numerical example that considers connectivity constraints in topology optimization problems, which proved the effectiveness of the proposed VSFM. Later, Liu and Wong [34] proposed a pairwise-structured RLR model to incorporate the prior knowledge of the relationship into weights between some features in an approximation way. However, the network-structured features and the connectivity between them have not been built in an appropriate model of feature selection.

To fully consider the sparsity, smoothness and connectivity in regularization, we established a connected network-regularized logistic regression (CNet-RLR) model for feature selection. We formulated CNet-RLR as an optimization model with network connectivity as constraints. In the regularization term, we firstly employed Lasso penalty to ensure the sparsity of regression coefficients, and graph penalty induced by the 2-Dirichlet form or graph Laplacian operator to ensure that adjacent nodes on a priori graph or network have similar coefficients, that is, smoothness. Node cut set and *Dirac* measure are used to give inequality constraints on the connectivity between features. We guaranteed the connectivity of selecting features when working on the feature network by algebraic connectivity.

Theoretically, we proved the grouping effect and oracle property of the CNet-RLR model. For overcoming the computational challenges produced by the non-differentiability of the Lasso penalty, we transformed it to be an equivalent formulation by introducing an auxiliary variable and design an interior-point algorithm to solve the transformed model. Specifically, a barrier problem is constructed with the help of the logarithmic barrier function, and Newton-Raphson method is employed to solve the Karush-Kuhn-Tucker (KKT) equation formed by the barrier problem. We also proved barrier convergence theorem and algorithm convergence theorem, which confirmed the feasibility and effectiveness of our proposed algorithm.

Finally, we validated CNet-RLR model in two experiments, one simulated dataset and one world-real cancer dataset of uterine corpus endometrial carcinoma (UCEC) respectively. The results of feature selection, biomarker identification, functional enrichment analysis and external dataset verification provided multiple evidence for the effectiveness of the proposed CNet-RLR. The comparison study between CNet-RLR and the other four RLR models and a graph embedded feature selection model demonstrated the advantages of our proposed method.

The major contributions of this paper are summarized as follows:

1. A connected network regularized logistic regression (CNet-RLR) model is proposed to select the features/variables in the form of a network with connectivity.
2. An inequality constraint induced by node cut set and *Dirac* measure is used to impose the connectivity between nodes during selecting features on a graph.
3. The proofs including the grouping effect and oracle property of the CNet-RLR model, the equivalence of the barrier problem with the original problem, and the convergence of the interior-point algorithm are given theoretically.
4. The validations of the presented CNet-RLR model are performed both on simulated datasets and real datasets. The comparisons with other representative models result in better performances of classification with more interpretable results.

The remainder of this paper is organized as follows. In Section 2, we review the related works for completeness and introduce our proposed CNet-RLR model in details. In Section 3, we present the barrier problem and interior-point algorithm for solving the CNet-RLR model. In Section 4, we prove the grouping effect, oracle property, and the convergence of the CNet-RLR model, respectively. In Section 5, we conduct experiments of the CNet-RLR model and compare its performances with the other five methods (namely, Lasso-RLR, Enet-RLR, Net-RLR, AbNet-RLR and GEDFN) on both simulation and real-world dataset about UCEC. Finally, we give the discussion and brief conclusions in Sections 6 and 7, respectively.

2 Connected network-regularized logistic regression (CNet-RLR)

2.1 Regularized logistic regression (RLR)

For completeness, we briefly introduce RLR firstly. Given a dataset $\mathcal{D}$ which includes n independent and identically distributed (i.i.d) observations

$$\mathcal{D} = \{(\boldsymbol{X}_1, y_1),\ (\boldsymbol{X}_2, y_2),\ \cdots,\ (\boldsymbol{X}_n, y_n)\},$$

where $\boldsymbol{X}_i = (x_{i1}, x_{i2}, \cdots, x_{ip})^{\mathrm{T}} \in \mathbb{R}^p$ denotes a p-dimensional sample vector, x_{ij} $(j = 1, 2 \cdots, p)$ represents the value of the j-th variable in the i-th sample. The corresponding variable $y_i \in \{0, 1\}$, where $i = 1, 2, \cdots, n$.

Originally, a logistic function is defined as follows

$$\pi_i = \Pr(y_i | \boldsymbol{X}_i; \boldsymbol{\theta}) = f\left(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta}\right) = \frac{\exp\left(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta}\right)}{1 + \exp\left(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta}\right)}, \quad (2.1)$$

where $f(\cdot)$ is the $Sigmoid$ function, $\boldsymbol{\theta} = \left(\theta_0, \theta_1, \ldots, \theta_p\right)^{\mathrm{T}}$ is a vector of coefficients, and θ_0 is the intercept term.

Taking logit transformation to both sides of (2.1) with respect to $\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta}$ gives the desired LR classifier for training and testing

$$\mathrm{logit}(\pi_i) = \log\frac{\pi_i}{1 - \pi_i} = \theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + \cdots + \theta_p x_{ip}. \quad (2.2)$$

The likelihood function associated with all samples is given by $\prod_{i=1}^{n} \pi_i$. Taking the log transform of bothsides, we have the log-likelihood function

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}|\mathcal{D}) &= \sum_{i=1}^{n} \log \Pr(y_i|\boldsymbol{X}_i;\boldsymbol{\theta}) \\ &= \sum_{i=1}^{n} \left\{ y_i \log\left[f\left(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta}\right)\right] + (1-y_i)\log\left[1-f\left(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta}\right)\right]\right\}. \quad (2.3)\end{aligned}$$

Clearly, (2.3) is a convex function of variable $\boldsymbol{\theta}$. We can determine the regression coefficient vector $\boldsymbol{\theta}$ by minimizing the negative log-likelihood function (2.3), i.e.,

$$\boldsymbol{\theta} = \arg\min\left\{-\mathcal{L}(\boldsymbol{\theta}|\mathcal{D})\right\}. \quad (2.4)$$

For conventional LR model, if the fitted model has many feature variables, and their corresponding regression coefficients are relatively large, that is, when $\boldsymbol{\theta}$ is large, Problem (2.4) is prone to over-fitting [54]. The commonly used standard method to avoid over-fitting is regularization [17, 55]. Its idea is to add an additional term that penalizes the large coefficients to the loss function of LR model to balance the objective function, thereby obtaining the regression coefficient $\boldsymbol{\theta}$ that makes the new objective function take the minimum value. If we choose an appropriate penalty for Problem (2.4), it becomes

$$\boldsymbol{\theta} = \arg\min\left\{-\mathcal{L}(\boldsymbol{\theta}|\mathcal{D}) + \mathcal{P}(\boldsymbol{\theta};\boldsymbol{\lambda})\right\}, \quad (2.5)$$

where $\mathcal{L}(\boldsymbol{\theta}|\mathcal{D})$ and $\mathcal{P}(\boldsymbol{\theta};\boldsymbol{\lambda})$ represent the loss function and the penalty function, respectively. In particular, $\boldsymbol{\lambda}$ is a vector of positive tuning parameter that controls the balance between them.

2.2 Lasso penalty

Lasso penalty has received much attention for its sparsity and convexity [56]. The regularization term of Lasso is defined as

$$\mathcal{P}(\boldsymbol{\theta};\lambda_1) = \lambda_1 \sum_{j=1}^{p} |\theta_j|, \quad (2.6)$$

where $\lambda_1 > 0$ is the regularization tuning parameter [20]. Here we call model (2.5) with penalty (2.6) Lasso-regularized logistic regression (Lasso-RLR) [15], which typically yields a sparse vector $\boldsymbol{\theta}$. When $\theta_i = 0$ $(i = 1, 2, \cdots, p)$, the Lasso-RLR model does not select the i-th feature [57], thus $\boldsymbol{\theta}$ uses only a few of selected features.

Given a set of objects $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_n]^{\mathrm{T}} \in \mathbb{R}^{n\times p}$ with labels $\boldsymbol{y} = (y_1, y_2, \cdots, y_n)^{\mathrm{T}} \in \mathbb{R}^n$. In view of the research background of actual problems, we usually assume that there are pairwise relationships between the variables under study. For many problems in which significant features are measured and represented on a graph or a network, neighboring/adjacent variables do not exist alone but are largely correlated. This makes the regression coefficients show some smoothness while maintaining sparsity [39].

2.3 Graph penalty

Graph penalty is a smoothness penalty based on a simple undirected graph, where the nodes stand for the data instances and the edges between nodes reflect their correlation and interaction. In this section, we give some notations, definitions and functions used in this paper.

2.3.1 Notations

A graph $G = (V, E)$ is composed of V and $E \subseteq V \times V$, where V is a finite set consisting of the vertices of G, E is a subset consisting of the edges of G. Self-loops refer to edges that connect the same starting vertex and end vertex of G, and parallel edges refer to two or more edges that connect the same pair of vertices [58]. Graph G is an undirected graph iff the set E has symmetry or adjoint property. In other words, if $e_{kj} \in E$, then $e_{jk} \in E$. The weighted undirected graph is defined as, for all $e_{kj} \in E$, $\forall\ k, j \in V$, there is a symmetric function w: $E \to \mathbb{R}^*$ satisfies $w(e_{kj}) = w(e_{jk})$. For simplicity, we denote $w(e_{kj})$ as w_{kj} in the case of without causing confusion.

A simple graph refers to the graph without self-loops or parallel edges in a weighted undirected graph where the weights are 0 or 1 [37]. For a simple graph, let $k \sim j$ be the edges formed by all vertices adjacent with vertex k, we may define its degree function $d : V \to \mathbb{R}^*$ as

$$d(k) := \sum_{k\sim j} w_{kj},$$

where

$$w_{kj} = \begin{cases} 1, & \text{if } k \text{ and } j \text{ are adjacent,} \\ 0, & \text{otherwise.} \end{cases}$$

2.3.2 Operators

Let $\mathcal{H}(V)$ represent the Hilbert space with the inner product operation $\langle f, g\rangle_{\mathcal{H}(V)} := \sum_{k\in V} f(k)g(k)$, for $\forall\ f, g \in \mathcal{H}(V)$. For ease of notations, we will omit the subscript of inner product, i.e., denote $\langle f, g\rangle_{\mathcal{H}(V)}$ as $\langle f, g\rangle$. The definition of $\mathcal{H}(E)$ is quite similar to that given earlier for $\mathcal{H}(V)$ and so is omitted. We firstly list the definitions of several gradient operators.

Definition 2.1 ***(Discrete gradient operator)*** [59] *The graph gradient ∇ is an operator from $\mathcal{H}(V)$ to $\mathcal{H}(E)$, it is*

defined to be

$$(\nabla f)(e_{kj}) := \sqrt{\frac{w_{jk}}{d_j}} f(j) - \sqrt{\frac{w_{kj}}{d_k}} f(k), \quad \forall\, e_{kj} \in E.$$

Definition 2.2 ***(p-Dirichlet form)*** [59] *For* $\forall\ f \in \mathcal{H}(V)$ *and* $\forall\ k \in V$, *the p-Dirichlet form* [60] *of the function f at vertex k is defined by*

$$\mathcal{R}_p(f) := \frac{1}{2} \sum_{k \in V} \|\nabla_k f\|^p, \tag{2.7}$$

where $\|\nabla_k f\| := \sqrt{\sum_{k \sim j} (\nabla f)^2 (e_{kj})}$ *represents the graph gradient norm.*

Intuitively, the discrete gradient operator $(\nabla f)(e_{kj})$ is used to measure the variation of f on edge e_{kj} of graph G, the graph gradient norm $\|\nabla_k f\|$ is developed to measure the roughness of f around vertex k of graph G. It is easy to show that p-Dirichlet form $\mathcal{R}_p(f)$ can be seen as the sum of the local variations caused by f between nearby vertexes of a graph, which can measure the whole roughness of f on graph G. As we have anticipated, $\mathcal{R}_p(f)$ can be used as a constraint operator to measure smoothness as long as it does not vary too much between nearby vertexes of graph G.

In particular, when $p = 2$, p-Dirichlet form (2.7) can be represented by the inner product

$$\mathcal{R}_2(f) := \frac{1}{2} \langle \nabla f, \nabla f \rangle. \tag{2.8}$$

Combining with Definition 2.1, the 2-Dirichlet form shown in (2.8) can be rewritten to the following formula

$$\mathcal{R}_2(f) := \frac{1}{2} \sum_{\substack{k \sim j \\ k \in V}} \left(\frac{f_k}{\sqrt{d_k}} - \frac{f_j}{\sqrt{d_j}} \right)^2 w_{kj}, \tag{2.9}$$

where $\sum_{\substack{k \sim j \\ k \in V}} \triangleq \sum_{k \in V} \sum_{k \sim j}$ denotes the sum over all unordered pairs $k \sim j$ for which k and j are adjacent.

Equation (2.9) indicates that we do not simply define the local variation on an edge by the difference of the function values on its two end points. The smoothness term essentially splits the function value at each point among the edges attached to it before computing the local changes, and the value assigned to each edge is proportional to its weight [61]. Mathematically, such a definition can make us finally recover the well-known graph Laplacian in a way parallel to continuous case [59].

2.3.3 Graph Laplacian

Given a dataset $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$, let $W \in \mathbb{R}^{p \times p}$ be the weight matrix of graph G. For a given prior network (e.g., RegNetwork for gene interactions [36]) of variables, define the degree matrix D with element d_k,

$$d_k = \begin{cases} \sum\limits_{k \sim j} w_{kj}, & \text{if } k \text{ and } j \ areadjacent, \\ 0, & \text{if } k \text{ is an isolated vertex,} \end{cases}$$

which represents the degree of vertex $k \in V$ ($k = 1, , \cdots, p$). Obviously, D is a diagonal matrix with non-diagonal elements of 0.

Laplacian matrix L [37] and its normalized version $\mathcal{L}$ [60] associated with graph G are introduced to illustrate that the p explanatory features are measured in a graph. For the Laplacian matrix, $L = \{l_{kj}\}$ with $k, j = 1, 2, \cdots, p$, where l_{kj} defined as follows

$$l_{kj} = \begin{cases} d_k - w_{kk}, & \text{if } k = j, \\ -w_{kj}, & \text{if } k \text{ and } j \text{ are adjacent,} \\ 0, & \text{otherwise.} \end{cases}$$

While for the normalized Laplacian matrix $\mathcal{L}$, it is made up of $\hat{l}_{kj}$ ($k, j = 1, 2, \cdots, p$) elements, which are defined by

$$\hat{l}_{kj} = \begin{cases} 1 - \dfrac{w_{kk}}{d_k}, & \text{if } k = j \text{ and } d_k \neq 0, \\ -\dfrac{w_{kj}}{\sqrt{d_k d_j}}, & \text{if } k \text{ and } j \text{ are adjacent,} \\ 0, & \text{otherwise.} \end{cases}$$

It is easy to verify that $\mathcal{L}$ is a symmetric positive semi-definite matrix with 0 as the smallest eigenvalue and 2 as the largest eigenvalue [39].

Based on simple algebra calculations, we can write $\mathcal{L}$ as

$$\mathcal{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}, \tag{2.10}$$

where I stands for the identity matrix, and the convention $D^{-1}(k, k) = 0$ for $d_k = 0$.

In order to establish the consistency between p-Dirichlet form and normalized graph Laplacian matrix $\mathcal{L}$, we give the following Lemma.

Lemma 2.1 ***(Cholesky factorization)*** [62] *If* $\mathcal{L} \in \mathbb{R}^{p \times p}$ *is a symmetric and positive definite matrix, then there exists a unique lower triangular* $S \in \mathbb{R}^{p \times m}$ *with positive diagonal entries such that* $\mathcal{L} = SS^T$, *where m is the total number of edges in graph G.*

Note that, by Lemma 2.1, $\mathcal{L}$ can be written as

$$\mathcal{L} = SS^{\mathrm{T}}, \tag{2.11}$$

where $S \in \mathbb{R}^{p \times m}$ is the matrix where the rows are indexed by the vertices and the columns are indexed by the edges of G such that each column corresponding to an edge $e = (k \sim j)$ has an entry $\frac{\sqrt{w_{kj}}}{d_k}$ in the row corresponding to k,

an entry $-\frac{\sqrt{w_{kj}}}{d_j}$ in the row corresponding to j, and zero entries elsewhere [37].

From the point of view of a graph structure, combining with p-Dirichlet form in Definition 2.2, (2.9) corresponds to graph Laplacian

$$\mathcal{R}_2(f) = \frac{1}{2} f^{\mathrm{T}} \mathcal{L} f, \tag{2.12}$$

where additional factor of $\frac{1}{2}$ in (2.12) is owing to that each edge is counted twice [59].

In this work, the smoothness of coefficient vector $\boldsymbol{\theta}$ with respect to the graph structure can be expressed as 2-Dirichlet form of $\boldsymbol{\theta}$. Combining with (2.12), we define the graph penalty as $\mathcal{P}(\boldsymbol{\theta}; \lambda_2) = \lambda_2 \mathcal{R}_2(\boldsymbol{\theta})$, namely,

$$\mathcal{P}(\boldsymbol{\theta}; \lambda_2) = \frac{\lambda_2}{2} \boldsymbol{\theta}^{\mathrm{T}} \mathcal{L} \boldsymbol{\theta}, \tag{2.13}$$

where $\lambda_2 > 0$ is another regularization parameter. In the sense of ignoring a constant factor, $\mathcal{R}_2(\boldsymbol{\theta})$ is equivalent to the form of graph Laplacian $\boldsymbol{\theta}^{\mathrm{T}} \mathcal{L} \boldsymbol{\theta}$ widely-used in literature [30, 33, 39, 47].

2.4 Algebraic connectivity

As a symmetric positive semi-definite matrix, the eigenvalues of normalized graph Laplacian matrix $\mathcal{L}$ are all non-negative real [40]. In fact, $\mathcal{L}$ always has at least one 0 eigenvalue with the corresponding eigenvector $\mathbf{1} = (1, 1, \cdots, 1)^{\mathrm{T}}$, and the second smallest eigenvalue, denoted by λ_2, is called the algebraic connectivity of graph G [40]. Thus if G is a connected graph (only consists of a single component) iff the second eigenvalue of $\mathcal{L}$ satisfies $\lambda_2 > 0$.

The following Theorem 2.1 is the representation for the second smallest eigenvalue of a symmetric matrix. Its proof can be found in Appendix A.1. Proof of Theorem 2.1.

Theorem 2.1 [40] *Let $\mathcal{L}$ be a real symmetric $n \times n$ matrix with eigenvalues $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$. Let $\boldsymbol{u} \in \mathbb{R}$ be an eigenvector of $\mathcal{L}$ corresponding to eigenvalue λ_1, then*

$$\lambda_2 = \min_{\substack{\boldsymbol{v} \neq \mathbf{0} \\ \boldsymbol{v} \perp \boldsymbol{u}}} \left\{ \frac{\boldsymbol{v}^T \mathcal{L} \boldsymbol{v}}{\boldsymbol{v}^T \boldsymbol{v}} \right\}, \tag{2.14}$$

where both $\boldsymbol{v}$ and $\mathbf{0}$ are vectors with the same dimension as vector $\boldsymbol{u}$. The minimum is taken over all non-zero vectors $\boldsymbol{v}$, orthogonal to $\boldsymbol{u}$.

2.5 Connectivity constraint

For a simple undirected graph $G = (V, E)$, considering the subset $C \subseteq V$. Let $C \subseteq V$ be the set of all vertices from all connected components of G. Define $G(C)$ to be the graph on C whose elements are precisely the connected components of G having both endpoints in C. G is connected if any two of its vertices are linked by a path in G [48]. Similarity, C is connected if $G(C)$ is connected. Define the set U is a connected component of C if $C \cup \{k\}$ is disconnected for all $k \in C \setminus U$ [63].

In many network applications, the aims are to identify a connected subset of vertices that exhibits desirable properties [63]. Node cut set of a graph provides the basis of imposing connectivity constraints in graph G [64], existing research work has shown that it can be innovatively used to enhance connectivity [63]. The following lists several definitions related to node cut set.

Definition 2.3 ***(kj-node cut set)*** *Given vertices k ($\in U$) and j ($\in U$) that are nonadjacent on network, i.e., $e_{kj} \notin E$, a set of nodes $S \subseteq U \setminus \{k, j\}$ is a node cut set separating k and j (or simply a kj-node cut set) if there is no path between k and j in $G[U \setminus S]$.*

Corollary 2.1 *Given $S \subseteq U$ and a nonadjacent pair of nodes $\{k, j\} \in U$, then there exists a path in $G(U)$ between k and j iff all kj-node cut sets S satisfy $S \cap U \neq \varnothing$.*

Definition 2.4 ***(Minimal kj-node cut set)*** *For $\forall\, e_{kj} \notin E$, define*

$$\Gamma(k, j) = \{S \subseteq U \setminus \{k, j\} : S \text{ is a minimal } kj\text{-node cut set}\},$$

then $\Gamma(kj)$ is called the minimal kj-node cut set.

Combine the above definitions, as shown in Fig. 1, every path in $G(U)$ between k and j intersects S, set S is a kj-node cut.

Here, we use the Algorithm 1 to find minimal kj-node cut set based on breadth-first search (BFS) algorithm, which is an algorithm to traverse a graph [65].

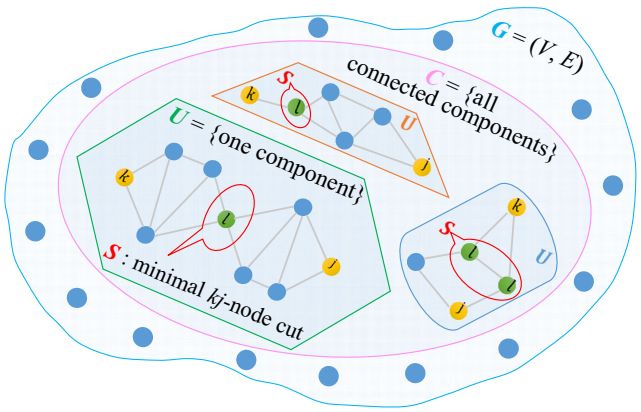

Fig. 1 The kj-node cut S in every connected component of a graph

Algorithm 1 The algorithm for finding minimal kj-node cut set.

Input: A graph $G = (V, E)$.
Output: Minimal kj-node cut set.

1: Calculate all connected components of graph G by using a simple BFS algorithm;
2: For each connected component, calculate its diameter by using a BFS-like algorithm, which searches a path with the actual diameter;
3: If there are more than one shortest paths of the length of diameter, choose the first one to be found, otherwise choose the only one;
4: Get two vertices connected by the diameter path from vertex k to vertex j;
5: For given vertex k and vertex j, find all minimal kj-node cut of each connected component [66].

In order to deduce the connectivity theorem that plays a very critical role in this paper, we introduce two key functions induced by the following two definitions.

Definition 2.5 ***(Characteristic function)*** [67] *Let C be the basic set. For the subset $U \subset C$, define the function*

$$\mathcal{F}(x) = \begin{cases} 1, & x \in U, \\ 0, & x \in C \setminus U, \end{cases}$$

on C. Then $\mathcal{F}(x)$ is called the characteristic function or indicator function of the set U relative to C. It is denoted as $\boldsymbol{\chi}_{U \subset C}(x)$, and abbreviated as $\boldsymbol{\chi}_U(x)$.

Definition 2.6 ***(Dirac measure)*** [67] *Let C be a nonempty set and let $x \in C$. For every $U \subseteq C$, define $\boldsymbol{\delta}_x(U) = \boldsymbol{\chi}_U(x)$, then $\boldsymbol{\delta}_x$ is called the Dirac measure at point x.*

Combining Definitions 2.5 and Definitions 2.6, it can easily find that $\mathcal{F}(x) = \boldsymbol{\chi}_U(x) = \boldsymbol{\delta}_x(U)$ holds for each vertex $x \in U$. Now, we consider the connectivity constraints of set U based on vertex variables and kj-node cut [51, 52, 63]. In particular, it can be concluded as Theorem 2.2.

Theorem 2.2 *To ensure set U be connected, the following systems of inequalities suffice,*

$$\sum_{l \in S} \boldsymbol{\delta}_l(U) \geqslant \boldsymbol{\delta}_k(U) + \boldsymbol{\delta}_j(U) - 1, \tag{2.15}$$

for $\forall\, S \in \Gamma(k, j),\ \forall\, k, j \in C,\ e_{kj} \notin E,\ \forall\, U \subseteq C \subseteq G$.

In constraints (2.15), we observe that if k and j are both selected, i.e., $\boldsymbol{\delta}_k(U) = \boldsymbol{\delta}_j(U) = 1$, then they force at least a node of each minimal kj-node cut S to be selected as well [48, 51, 52, 63]. Thus for the model (2.5), we achieved that if both coefficients θ_k and θ_j (corresponding to vertexes k and j) are all non-zeros, then θ_l (corresponding to vertex l) is also non-zero.

2.6 CNet-RLR

Considering the sparsity and smoothness of the coefficient $\boldsymbol{\theta}$, as well as the connectivity of a network, we propose a connected network-regularized penalty $\mathcal{P}(\boldsymbol{\theta}; \boldsymbol{\lambda})$, which is a combination of Lasso penalty (2.6), graph penalty (2.13) and connectivity constraints (2.15). Namely,

$$\mathcal{P}(\boldsymbol{\theta}; \boldsymbol{\lambda}) = \mathcal{P}_{Lasso}(\boldsymbol{\theta}; \lambda_1) + \mathcal{P}_{Graph}(\boldsymbol{\theta}; \lambda_2), \tag{2.16}$$

where $\mathcal{P}_{Lasso}(\boldsymbol{\theta}; \lambda_1) = \lambda_1 \sum_{j=1}^{p} |\theta_j|$, $\mathcal{P}_{Graph}(\boldsymbol{\theta}; \lambda_2) = \frac{\lambda_2}{2} \boldsymbol{\theta}^{\mathrm{T}} \mathcal{L} \boldsymbol{\theta}$, and

$$\sum_{l \in S} \boldsymbol{\delta}_l(U) \geqslant \boldsymbol{\delta}_k(U) + \boldsymbol{\delta}_j(U) - 1, \tag{2.17}$$

for $\forall\, S \in \Gamma(k, j),\ \forall\, k, j \in U,\ e_{kj} \notin E,\ \forall\, U \subseteq C \subseteq G$.

For Penalty (2.16), the Lasso penalty generates a sparse solution, and the 2-Dirichlet form induces a smoothness solution. For Constraints (2.17), the inequalities guarantee the connectivity between vertices/variables. Based on these key points, Problem (2.5) combined with Penalty (2.16) and Constraints (2.17) constitutes a connected network-regularized logistic regression (CNet-RLR) model

$$\begin{aligned} &\text{Minimize} \quad -\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) + \lambda_1 \sum_{j=1}^{p} |\theta_j| + \frac{\lambda_2}{2} \boldsymbol{\theta}^{\mathrm{T}} \mathcal{L} \boldsymbol{\theta} \\ &\text{Subject to} \quad \boldsymbol{\delta}_k(U) + \boldsymbol{\delta}_j(U) - 1 \leq \sum_{l \in S} \boldsymbol{\delta}_l(U), \\ &\forall\, S \in \Gamma(k, j),\ \forall\, k, j \in U,\ e_{kj} \notin E,\ \forall U \subseteq C \subseteq G. \end{aligned} \tag{2.18}$$

(2.18) is a constrained optimization problem, in which $\mathcal{L}(\boldsymbol{\theta}; \mathcal{D})$ is defined by (2.3).

3 Interior-point algorithm

3.1 Equivalent problem

The Lasso penalty is a convex function but isn't differentiable specifically when all coefficients are 0s, which brings computational challenges to get its solution [57]. A feasible approach is to transform the problem into one with differentiable objective and constraint functions by introducing an auxiliary variable. In this section, let auxiliary variable

$\boldsymbol{u} = (u_1, u_2, \cdots, u_p)^{\mathrm{T}} \in \mathbb{R}^p$, we derive an equivalent problem with linear inequality constraints of Problem (2.18)

$$\begin{aligned} \text{Minimize} \quad & -\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) + \lambda_1 \sum_{j=1}^{p} u_j + \frac{\lambda_2}{2} \boldsymbol{\theta}^{\mathrm{T}} \mathcal{L} \boldsymbol{\theta} \\ \text{Subject to} \quad & u_j - \theta_j \geq 0, \quad j = 1, 2, \cdots, p, \\ & u_j + \theta_j \geq 0, \quad j = 1, 2, \cdots, p, \\ & \sum_{l \in S} \boldsymbol{\delta}_l(U) \geq \boldsymbol{\delta}_k(U) + \boldsymbol{\delta}_j(U) - 1, \\ & \forall\, S \in \Gamma(k, j),\ \forall\, k, j \in U,\ e_{kj} \notin E,\ \forall\, U \subseteq C \subseteq G, \end{aligned} \tag{3.1}$$

where u_j $(j = 1, ..., p)$ denotes the j-th component of vector $\boldsymbol{u} \in \mathbb{R}^p$.

For the equivalence with Problem (2.18), we note that $u_j = |\theta_j|$ must hold at the optimal point for Problem (3.1), in which case the objective functions in these two problems are the same [57]. The equivalent Problem (3.1) is differentiable for all positive u_j of $\boldsymbol{u}$ [68]. It is a convex optimization problem with a smooth objective function and some linear constraint conditions, we can solve it by means of the technique of interior-point [69], one of the standard convex optimization solving methods [57].

3.2 Logarithmic barrier function

Considering a general inequality constraints problem

$$\begin{aligned} \text{Minimize} \quad & f(x) \\ \text{Subject to} \quad & x \geq 0, \end{aligned} \tag{3.2}$$

where $x \in \mathbb{R}^n$. After making the natural log barrier term for inequality constraints, we have

$$\phi(x) = -\mu \log(x), \tag{3.3}$$

where $\mu > 0$ is a penalty parameter. (3.3) is called the logarithmic barrier function, whose function curves under different values of μ are shown in Fig. 2. It can be seen that $\phi(x)$ is a function like a wall, whose value is zero when x does not violate the constraint, but tends to be positive infinity once x violates the constraint. The smaller μ, the better the approximation effect of $\phi(x)$, and when $\mu = 0$, $\phi(x) = 0$.

Augmenting the objective function of Problem (3.2) by the logarithmic barrier function, we get

$$\text{minimize} \quad f(x) - \mu \sum_{i=1}^{n} \log(x_i), \tag{3.4}$$

where the term $\log(x_i)$ is undefined for $x_i < 0$. Thus, the Problem (3.4) can search points only in the interior feasible domain.

3.3 Barrier problem

Here, for the linear inequality constraints of Problem (3.1), we define its logarithmic barrier function [68, 70] as

$$\begin{aligned} \phi_\mu(\boldsymbol{\theta}, \boldsymbol{u}) &= -\mu \sum_{j=1}^{p} \log\left(u_j - \theta_j\right) - \mu \sum_{j=1}^{p} \log\left(u_j + \theta_j\right) \\ &= -\mu \sum_{j=1}^{p} \log\left(u_j^2 - \theta_j^2\right), \end{aligned} \tag{3.5}$$

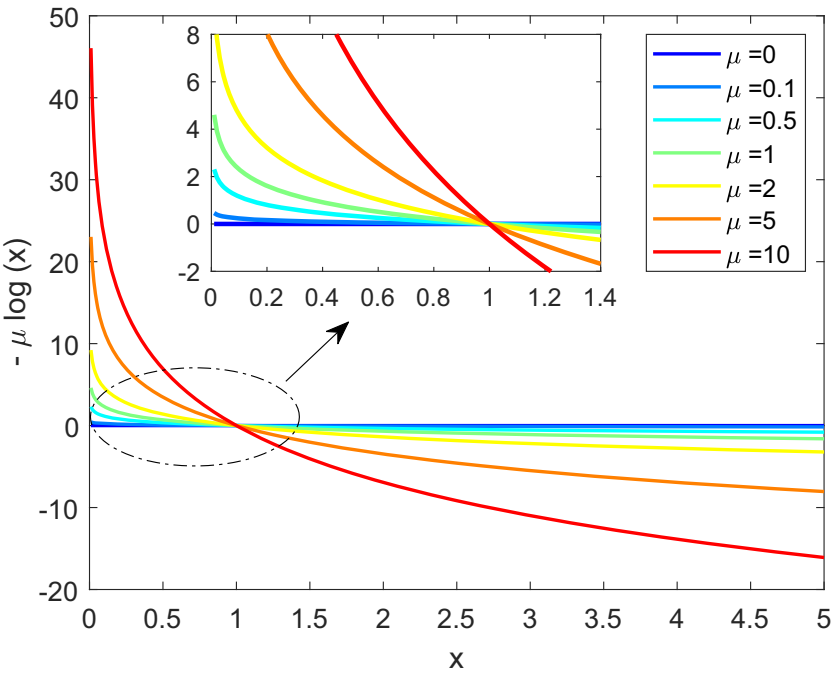

Fig. 2 The function curves of logarithmic barrier function under different values of μ

with the non-empty solution domain

$$\operatorname{dom}\phi_\mu = \left\{(\boldsymbol{\theta}, \boldsymbol{u}) \in \mathbb{R}^p \times \mathbb{R}^p \ : \ |\theta_j| < u_j, \ j = 1, 2, \cdots, p\right\}.$$

Note that (3.5) is smooth and convex. Augmenting the objective function of Problem (3.1) by the logarithmic barrier function (3.5), we obtain a barrier problem

$$\begin{aligned} \text{Minimize} \quad & -\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) + \lambda_1 \sum_{j=1}^{p} u_j + \frac{\lambda_2}{2}\boldsymbol{\theta}^{\mathrm{T}}\mathcal{L}\boldsymbol{\theta} - \mu \sum_{j=1}^{p} \log\left(u_j^2 - \theta_j^2\right) \\ \text{Subject to} \quad & \sum_{l \in S} \boldsymbol{\delta}_l(U) \geq \boldsymbol{\delta}_k(U) + \boldsymbol{\delta}_j(U) - 1, \\ & \forall\, S \in \Gamma(k, j),\ \forall\, k, j \in U,\ e_{kj} \notin E,\ \forall U \subseteq C \subseteq G. \end{aligned} \tag{3.6}$$

for a decreasing sequence of barrier parameters μ converging to zero

$$\mu_{j+1} = \beta \mu_j. \tag{3.7}$$

Let

$$Q_\mu(\boldsymbol{\theta}, \boldsymbol{u}) = -\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) + \lambda_1 \mathbf{1}^{\mathrm{T}}\boldsymbol{u} + \frac{\lambda_2}{2}\boldsymbol{\theta}^{\mathrm{T}}\mathcal{L}\boldsymbol{\theta} - \mu \sum_{j=1}^{p} \log\left(u_j^2 - \theta_j^2\right)$$

be the objective function of Problem (3.6), where $\mathbf{1} \in \mathbb{R}^p$ represents a column vector whose components are all one. Clearly, $Q_\mu(\boldsymbol{\theta}, \boldsymbol{u})$ is smooth and strictly convex. So it has a unique minimizer, we denote it as $(\boldsymbol{\theta}^*, \boldsymbol{u}^*)$.

3.4 Karush-Kuhn-Tucker (KKT) conditions

Let

$$z_j^\theta = \frac{2\theta_j \mu}{u_j^2 - \theta_j^2}, \quad z_j^u = \frac{2u_j \mu}{u_j^2 - \theta_j^2}, \quad j = 1, 2, \cdots, p, \tag{3.8}$$

then the explicit equations for the gradient of $Q_\mu(\boldsymbol{\theta}, \boldsymbol{u})$ is given by

$$\nabla Q_\mu(\boldsymbol{\theta}, \boldsymbol{u}) = \begin{bmatrix} \dfrac{\partial Q_\mu}{\partial \boldsymbol{\theta}} \\ \dfrac{\partial Q_\mu}{\partial \boldsymbol{u}} \end{bmatrix} \in \mathbb{R}^{2p}, \tag{3.9}$$

where

$$\frac{\partial Q_\mu}{\partial \boldsymbol{\theta}} = \boldsymbol{X}^{\mathrm{T}}(\boldsymbol{f} - \boldsymbol{y}) + \lambda_2 \mathcal{L}\boldsymbol{\theta} + \boldsymbol{z}^\theta \in \mathbb{R}^p, \quad \frac{\partial Q_\mu}{\partial \boldsymbol{u}} = \lambda_1 \mathbf{1} - \boldsymbol{z}^u \in \mathbb{R}^p,$$

with $\boldsymbol{z}^\theta = (z_1^\theta, z_2^\theta, \cdots, z_p^\theta)^{\mathrm{T}}$, $\boldsymbol{z}^u = (z_1^u, z_2^u, \cdots, z_p^u)^{\mathrm{T}}$.

The KKT conditions for the barrier problem (3.6) can be written as

$$\begin{cases} \boldsymbol{X}^{\mathrm{T}}(\boldsymbol{f} - \boldsymbol{y}) + \lambda_2 \mathcal{L}\boldsymbol{\theta} + \boldsymbol{z}^\theta = 0, \\ \lambda_1 \mathbf{1} - \boldsymbol{z}^u = 0, \\ Z^\theta \Sigma \mathbf{1} - 2\mu\boldsymbol{\theta} = 0, \\ Z^u \Sigma \mathbf{1} - 2\mu\boldsymbol{u} = 0. \end{cases} \tag{3.10}$$

where

$$\Sigma = \begin{bmatrix} u_1^2 - \theta_1^2 & & & \\ & u_2^2 - \theta_2^2 & & \\ & & \ddots & \\ & & & u_p^2 - \theta_p^2 \end{bmatrix}, Z^\theta = \begin{bmatrix} z_1^\theta & & & \\ & z_2^\theta & & \\ & & \ddots & \\ & & & z_p^\theta \end{bmatrix},$$

$$Z^u = \begin{bmatrix} z_1^u & & & \\ & z_2^u & & \\ & & \ddots & \\ & & & z_p^u \end{bmatrix}.$$

3.5 Newton-Raphson method

In order to obtain the KKT solutions for a given fixed barrier parameter μ_j, a Newton-Raphson method is applied to solve the (3.10). After simple calculations, we have

$$\begin{aligned} -\nabla_\theta^2 \mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) &= -\nabla_\theta \left(\sum_{i=1}^{n} \boldsymbol{X}_i \left\{ y_i - f(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta}) \right\} \right) \\ &= -\sum_{i=1}^{n} \left\{ \boldsymbol{X}_i - f'_{\boldsymbol{\theta}}(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta}) \right\} \\ &= \sum_{i=1}^{n} \left\{ \boldsymbol{X}_i f(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta}) \left(1 - f(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta})\right) \boldsymbol{X}_i^{\mathrm{T}} \right\} \\ &= \boldsymbol{X}^{\mathrm{T}} \hat{\Lambda} \boldsymbol{X}, \end{aligned}$$

where

$$\hat{\Lambda} = \begin{bmatrix} f(\boldsymbol{X}_1^{\mathrm{T}}\boldsymbol{\theta}) \cdot (1 - f(\boldsymbol{X}_1^{\mathrm{T}}\boldsymbol{\theta}) & & & \\ & f(\boldsymbol{X}_2^{\mathrm{T}}\boldsymbol{\theta}) \cdot (1 - f(\boldsymbol{X}_2^{\mathrm{T}}\boldsymbol{\theta}) & & \\ & & \ddots & \\ & & & f(\boldsymbol{X}_n^{\mathrm{T}}\boldsymbol{\theta}) \cdot (1 - f(\boldsymbol{X}_n^{\mathrm{T}}\boldsymbol{\theta}) \end{bmatrix},$$

Combining with (3.8), we have

$$\frac{\partial z_j^\theta}{\partial \theta} = \frac{2\mu(u_j^2+\theta_j^2)}{(u_j^2-\theta_j^2)^2}, \quad \frac{\partial z_j^\theta}{\partial u} = \frac{-4u_j\theta_j}{(u_j^2-\theta_j^2)^2}.$$

Similarly, we can prove that $\frac{\partial z_j^u}{\partial \theta} = -\frac{\partial z_j^\theta}{\partial u}$, $\frac{\partial z_j^u}{\partial u} = -\frac{\partial z_j^\theta}{\partial \theta}$.

In addition, we define

$${}^1D^\theta = \begin{bmatrix} \frac{\partial z_1^\theta}{\partial \theta} & & & \\ & \frac{\partial z_2^\theta}{\partial \theta} & & \\ & & \ddots & \\ & & & \frac{\partial z_p^\theta}{\partial \theta} \end{bmatrix}, \quad {}^2D^\theta = \begin{bmatrix} \frac{\partial z_1^\theta}{\partial u} & & & \\ & \frac{\partial z_2^\theta}{\partial u} & & \\ & & \ddots & \\ & & & \frac{\partial z_p^\theta}{\partial u} \end{bmatrix},$$

then it is easy to show that

$${}^1D^u = \begin{bmatrix} \frac{\partial z_1^u}{\partial \theta} & & & \\ & \frac{\partial z_2^u}{\partial \theta} & & \\ & & \ddots & \\ & & & \frac{\partial z_p^u}{\partial \theta} \end{bmatrix} = -{}^2D^\theta, \quad {}^2D^u = \begin{bmatrix} \frac{\partial z_1^u}{\partial u} & & & \\ & \frac{\partial z_2^u}{\partial u} & & \\ & & \ddots & \\ & & & \frac{\partial z_p^u}{\partial u} \end{bmatrix} = -{}^1D^\theta.$$

For convenience, we make

$${}^1\Omega^\theta = \frac{\partial\left(Z^\theta \Sigma\right)}{\partial \theta} = \begin{bmatrix} -2z_1^\theta\theta_1 & & & \\ & -2z_2^\theta\theta_2 & & \\ & & \ddots & \\ & & & -2z_p^\theta\theta_p \end{bmatrix},$$

$${}^2\Omega^\theta = \frac{\partial\left(Z^\theta \Sigma\right)}{\partial u} = \begin{bmatrix} 2z_1^\theta u_1 & & & \\ & 2z_2^\theta u_2 & & \\ & & \ddots & \\ & & & 2z_p^\theta u_p \end{bmatrix}.$$

Similarly, we can prove that

$${}^1\Omega^u = \frac{\partial\left(Z^u \Sigma\right)}{\partial \theta} = \begin{bmatrix} -2z_1^u\theta_1 & & & \\ & -2z_2^u\theta_2 & & \\ & & \ddots & \\ & & & -2z_p^u\theta_p \end{bmatrix},$$

$${}^2\Omega^u = \frac{\partial\left(Z^u \Sigma\right)}{\partial u} = \begin{bmatrix} 2z_1^u u_1 & & & \\ & 2z_2^u u_2 & & \\ & & \ddots & \\ & & & 2z_p^u u_p \end{bmatrix}.$$

Here, k is used to denote the iteration counter in the 'inner loop'. For any given μ_j, given an iterating $(\boldsymbol{\theta}_k, \boldsymbol{\mu}_k, z_k^\theta, z_k^u)$, the search direction $(\Delta\boldsymbol{\theta}_k, \Delta\boldsymbol{u}_k, \Delta z_k^\theta, \Delta z_k^u)$ is defined by the following Newton system

$$\begin{bmatrix} W_k + {}^1D_k^\theta & {}^2D_k^\theta & I & 0 \\ {}^2D_k^\theta & {}^1D_k^\theta & 0 & -I \\ {}^1\Omega_k^\theta - 2\mu_j I & {}^2\Omega_k^\theta & \Sigma_k & 0 \\ {}^1\Omega_k^u & {}^2\Omega_k^u - 2\mu_j I & 0 & \Sigma_k \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{\theta}_k \\ \Delta\boldsymbol{u}_k \\ \Delta z_k^\theta \\ \Delta z_k^u \end{bmatrix} = -\begin{bmatrix} X_k^{\mathrm{T}}(\boldsymbol{f}_k - \boldsymbol{y}_k) + \lambda_2\mathcal{L}_k\boldsymbol{\theta}_k + z_k^\theta \\ \lambda_1\mathbf{1} - z_k^u \\ Z_k^\theta\Sigma_k\mathbf{1} - 2\mu_j\boldsymbol{\theta}_k \\ Z_k^u\Sigma_k\mathbf{1} - 2\mu_j\boldsymbol{u}_k \end{bmatrix}, \tag{3.11}$$

where $W_k = \boldsymbol{X}_k^{\mathrm{T}}\hat{\Lambda}_k\boldsymbol{X}_k - \lambda_2\mathcal{L}_k \in \mathbb{R}^{p\times p}$, and $I \in \mathbb{R}^{p\times p}$ represents the identity matrix.

Rewrite (3.11), we have

$$\begin{cases} (W_k + {}^1D_k^\theta)\Delta\boldsymbol{\theta}_k + {}^2D_k^\theta\Delta\boldsymbol{u}_k + X_k^{\mathrm{T}}(\boldsymbol{f}_k - \boldsymbol{y}_k) + \lambda_2\mathcal{L}_k\boldsymbol{\theta}_k + (z_k^\theta + \Delta z_k^\theta) = 0, \\ {}^2D_k^\theta\Delta\boldsymbol{\theta}_k + {}^1D_k^\theta\Delta\boldsymbol{u}_k + \lambda_1\mathbf{1} - (z_k^u + \Delta z_k^u) = 0, \\ ({}^1\Omega_k^\theta - 2\mu_j I)\Delta\boldsymbol{\theta}_k + {}^2\Omega_k^\theta\Delta\boldsymbol{u}_k + \Sigma_k\Delta z_k^\theta + Z_k^\theta\Sigma_k\mathbf{1} - 2\mu_j\boldsymbol{\theta}_k = 0, \\ {}^1\Omega_k^u\Delta\boldsymbol{\theta}_k + ({}^2\Omega_k^u - 2\mu_j I)\Delta\boldsymbol{u}_k + \Sigma_k\Delta z_k^u + Z_k^u\Sigma_k\mathbf{1} - 2\mu_j\boldsymbol{u}_k = 0. \end{cases} \tag{3.12}$$

From the latter two equations of (3.12), we obtain

$$\begin{cases} \Delta z_k^\theta = -z_k^\theta + 2\mu_j\Sigma_k^{-1}\boldsymbol{\theta}_k - \Sigma_k^{-1}\left({}^1\Omega_k^\theta - 2\mu_j I\right)\Delta\boldsymbol{\theta}_k - \Sigma_k^{-1}\,{}^2\Omega_k^\theta\Delta\boldsymbol{u}_k, \\ \Delta z_k^u = -z_k^u + 2\mu_j\Sigma_k^{-1}\boldsymbol{u}_k - \Sigma_k^{-1}\,{}^1\Omega_k^u\Delta\boldsymbol{\theta}_k - \Sigma_k^{-1}\left({}^2\Omega_k^u - 2\mu_j I\right)\Delta\boldsymbol{u}_k. \end{cases} \tag{3.13}$$

Once the vectors Δz_k^θ and Δz_k^u are solved in (3.13), submitting them into the former two equations of (3.12), the vectors $\Delta\boldsymbol{\theta}_k$ and $\Delta\boldsymbol{u}_k$ are then obtained by the reduced Newton system:

$$\begin{bmatrix} W_k + {}^1D_k^\theta - \Sigma_k^{-1}({}^1\Omega_k^\theta - 2\mu I) & {}^2D_k^\theta - \Sigma_k^{-1}\,{}^2\Omega_k^\theta \\ {}^2D_k^\theta + \Sigma_k^{-1}\,{}^1\Omega_k^u & {}^1D_k^\theta + \Sigma_k^{-1}({}^2\Omega_k^u - 2\mu I) \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{\theta}_k \\ \Delta\boldsymbol{u}_k \end{bmatrix} = -\begin{bmatrix} X^{\mathrm{T}}(\boldsymbol{f}_k - \boldsymbol{y}) + \lambda_2\mathcal{L}\boldsymbol{\theta}_k + 2\mu_j\Sigma_k^{-1}\boldsymbol{\theta}_k \\ \lambda_1\mathbf{1} - 2\mu_j\Sigma_k^{-1}\boldsymbol{u}_k \end{bmatrix}. \tag{3.14}$$

When search directions are computed from (3.13) and (3.14), step sizes $\alpha_k, \alpha_k^\theta, \alpha_k^u \in (0, 1]$ have to be determined in order to obtain the next iteration as

$$\begin{cases} \boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \alpha_k\Delta\boldsymbol{\theta}_k, \\ \boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \alpha_k\Delta\boldsymbol{u}_k, \\ z_{k+1}^\theta = z_k^\theta + \alpha_k^\theta\Delta z_k^\theta, \\ z_{k+1}^u = z_k^u + \alpha_k^u\Delta z_k^u. \end{cases} \tag{3.15}$$

If $\alpha_k = \alpha_k^\theta = \alpha_k^\theta = 1$, (3.15) is called a full Newton step.

When the barrier parameter μ is fixed, the presented method can get the approximate solution of Problem (3.6). Decreases the barrier parameters gradually, then the

approximate solution of each new barrier problem (3.6) can be obtained on the basis of the approximate solution of the previous obstacle one [71]. Considering the individual parts of the KKT conditions shown in (3.10), the optimal error for Problem (3.6) can be defined as

$$\begin{cases} \left\| X^{\mathrm{T}}(\boldsymbol{f}-\boldsymbol{y}) + \lambda_2 \mathcal{L}\boldsymbol{\theta} + z^{\theta} \right\|_{\infty} \leqslant \epsilon_{\mu}, \\ \left\| \lambda_1 \mathbf{1} - z^{u} \right\|_{\infty} \leqslant \epsilon_{\mu}, \\ \left\| Z^{\theta} \Sigma \mathbf{1} - 2\mu\boldsymbol{\theta} \right\|_{\infty} \leqslant \epsilon_{\mu}, \\ \left\| Z^{u} \Sigma \mathbf{1} - 2\mu\boldsymbol{u} \right\|_{\infty} \leqslant \epsilon_{\mu}, \end{cases} \tag{3.16}$$

where $\|\boldsymbol{x}\|_{\infty} = \max_i |x_i|$ for any $\boldsymbol{x} = (x_1, x_2, \cdots, x_m)^{\mathrm{T}} \in \mathbb{R}^m$, and ϵ_{μ} is a precision parameter. It is related to the current penalty parameter μ and satisfies that when $\mu \to 0$, ϵ_{μ} decreases to 0 holds [70].

For simplicity, define

$$\begin{aligned} \mathrm{Res}_{\mu}(\boldsymbol{\theta}, \boldsymbol{u}, z^{\theta}, z^{u}) = \max \big\{ & \|X^{\mathrm{T}}(\boldsymbol{f}-\boldsymbol{y}) + \lambda_2 \mathcal{L}\theta + z^{\theta}\|_{\infty}, \|\lambda_1 \mathbf{1} - z^{u}\|_{\infty}, \\ & \|Z^{\theta}\Sigma\mathbf{1} - 2\mu\theta\|_{\infty}, \|Z^{u}\Sigma\mathbf{1} - 2\mu u\|_{\infty} \big\}, \end{aligned} \tag{3.17}$$

Thus (3.16) can be written as

$$\mathrm{Res}_{\mu}(\boldsymbol{\theta}, \boldsymbol{u}, z^{\theta}, z^{u}) \leqslant \epsilon_{\mu}, \tag{3.18}$$

where

$$\epsilon_{\mu_{j+1}} = \beta \epsilon_{\mu_j} \tag{3.19}$$

is a decreasing sequence of error parameters converging to zero in response to μ_j.

Defining (3.18) as $\mathrm{Res}_0(\boldsymbol{\theta}, \boldsymbol{u}, z^{\theta}, z^{u})$ when $\mu = 0$, it measures the optimality error for the equivalent problem (3.1). The iterative process stops with the approximate solution $(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{u}}, \tilde{z^{\theta}}, \tilde{z^{u}})$ converging to the optimal solution when KKT conditions are satisfied with a predefined positive error tolerance ϵ_{tol},

$$\mathrm{Res}_0(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{u}}, \tilde{z^{\theta}}, \tilde{z^{u}}) \leqslant \epsilon_{\mathrm{tol}}. \tag{3.20}$$

To sum up, in order to solve Problem (3.1), we propose the interior-point algorithm, that is, Algorithm 2.

Algorithm 2 Interior-point algorithm for solving the equivalent problem (3.1).

Input: Training dataset $\mathcal{D} = \{\boldsymbol{X}, \boldsymbol{y}\}$, Laplacian matrix $\mathscr{L}$, barrier parameter $\mu_0 > 0$, tolerance parameters $\epsilon_{\mu_0} > 0$ and $\epsilon_{\mathrm{tol}} > 0$, line search parameters $\alpha \in (0, 1)$ and $\beta \in (0, \frac{1}{2})$.

Output: $\boldsymbol{\theta}$, $\boldsymbol{u}$.

1: Initialize $\boldsymbol{\theta} = \mathbf{0}$, $\boldsymbol{u} = \mathbf{1}$, $k = 0$, $j = 0$;
2: **Repeat**
3: Compute the search direction $(\Delta\boldsymbol{\theta}_k, \Delta\boldsymbol{u}_k, \Delta z_k^{\theta}, \Delta z_k^{u})$ by solving (3.13) and (3.14);
4: Backtrack line search to find the smallest integer $k \geq 0$ that satisfies $Q_{\mu}(\boldsymbol{\theta} + \alpha^k \triangle\boldsymbol{\theta}, \boldsymbol{u} + \alpha^k \triangle\boldsymbol{u}) \leq Q_{\mu}(\boldsymbol{\theta}, \boldsymbol{u}) + \beta\alpha^k \nabla Q_{\mu}(\boldsymbol{\theta}, \boldsymbol{u})^{\mathrm{T}} [\triangle\boldsymbol{\theta}; \triangle\boldsymbol{u}]$, where α_k are based on Armijo linear search [71];
5: Update $(\boldsymbol{\theta}_k, \boldsymbol{\mu}_k, z_k^{\theta}, z_k^{u})$ using (3.15);
6: Let $k = k + 1$;
7: Compute the criteria (3.18) for testing convergence with ϵ_{μ_j};
8: Update μ_j by (3.7), ϵ_{μ_j} by (3.19);
9: Let $j = j + 1$;
10: **Until** the termination condition (3.20) is met with the tolerance ϵ_{tol};
11: Return $\boldsymbol{\theta}$, $\boldsymbol{u}$.

4 Theoretical properties

4.1 Grouping effect

Theorem 4.1 *Suppose that $\hat{\boldsymbol{\theta}}$ is estimated by solving the problem (2.18) on a given dataset $\mathcal{D}$, i.e.,*

$$\hat{\boldsymbol{\theta}} = \arg\min \Big\{ -\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) + \lambda_1 \sum_{j=1}^{p} |\theta_j| + \frac{\lambda_2}{2} \boldsymbol{\theta}^T \mathcal{L} \boldsymbol{\theta} \Big\}. \tag{4.1}$$

Define $\hat{\boldsymbol{X}}_i = (x_{1i}, x_{2i}, \cdots, x_{ni})^T \in \mathbb{R}^n$ $(i = 1, 2, \cdots, p)$ is the i-th column of the matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$. If $\hat{\boldsymbol{X}}_k = \hat{\boldsymbol{X}}_j$, then for any non-negative regularization parameters λ_1 and λ_2, it holds $\hat{\theta}_k = \hat{\theta}_j$.

Theorem 4.1 qualitatively illustrates the fact that if two variables/features are correlated in the network, their coefficients will be the same. The proof of this theorem can be found in Appendix A.2. Proof of Theorem 4.1.

Theorem 4.2 *Given dataset $\mathcal{D}$ with*

$$\sum_{i=1}^{n} x_{ij} = 0, \quad \sum_{i=1}^{n} x_{ij}^2 = 1, \quad \sum_{i=1}^{n} y_i = 0, \quad j = 1, 2, \cdots, p. \tag{4.2}$$

For any non-negative regularization parameters λ_1 and λ_2, let $\hat{\boldsymbol{\theta}}(\lambda_1, \lambda_2)$ be the solution of Problem (2.18). Assume

that $\hat{\theta}_k(\lambda_1,\lambda_2)\hat{\theta}_j(\lambda_1,\lambda_2) > 0$, $e_{kj} = 1$, *and* $d_k = d_j = w_{kj}$, *i.e., vertices* k *and* j *are only linked to each other in the network. Define*

$$D_{\lambda_1,\lambda_2}(k,j) = \frac{|\hat{\theta}_k(\lambda_1,\lambda_2) - \hat{\theta}_j(\lambda_1,\lambda_2)|}{\|\boldsymbol{y}\|_1}, \tag{4.3}$$

where $\|\boldsymbol{y}\|_1 = \sqrt{\sum_{i=1}^{n}|y_i|}$. *Then we have*

$$D_{\lambda_1,\lambda_2}(k,j) \le \frac{c\sqrt{2(1-\rho)}}{\lambda_2}, \tag{4.4}$$

where c *is a constant,* ρ *represents the variables/features correlation of* $\hat{\boldsymbol{X}}_k$ *and* $\hat{\boldsymbol{X}}_j$.

Equation (4.4) in Theorem 4.2 gives an upper bound for the grouping effect of the CNet-RLR model. If vertices k and j are highly correlated in a network, then their correlation $\rho = 1$, so it holds $D_{\lambda_1,\lambda_2}(k,j) = 0$ by (4.4) [31]. It means that for highly correlated features in a network, the difference between their estimated coefficients of genes k and j is almost 0 [30]. The proof of this theorem can be found in Appendix A.3. Proof of Theorem 4.2.

4.2 Oracle property

For proving the oracle property of CNet-RLR estimation, three regularization conditions are firstly illustrated. Assume that the real model can be sparsely represented, and the real parameters are recorded as $\bar{\boldsymbol{\theta}} = (\bar{\theta}_1, \bar{\theta}_2, \cdots, \bar{\theta}_p)^{\mathrm{T}}$. Without loss of generality, suppose all the first l components are not 0, and the last $p-l$ components are 0, that is $\bar{\boldsymbol{\theta}} = (\bar{\boldsymbol{\theta}}_1, \bar{\boldsymbol{\theta}}_2)^{\mathrm{T}}$, $\bar{\boldsymbol{\theta}}_1 \neq \mathbf{0}$, $\bar{\boldsymbol{\theta}}_2 = \mathbf{0}$.

The following conditions are called regularization conditions:

(1) The observation value $\boldsymbol{X}_i$ is i.i.d, and its probability density function is denoted as $f(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta})$, and abbreviated as $\boldsymbol{f}$, then the first-order derivative and the second-order derivative of the density function satisfy

$$E_{\boldsymbol{\theta}}\left[\frac{\partial \log \boldsymbol{f}}{\partial \theta_j}\right] = 0, \quad j = 1,2,\cdots,p, \tag{4.5}$$

$$I_{ij} = E_{\boldsymbol{\theta}}\left[\frac{\partial \log \boldsymbol{f}}{\partial \theta_i}\cdot\frac{\partial \log \boldsymbol{f}}{\partial \theta_j}\right] = E_{\boldsymbol{\theta}}\left[-\frac{\partial^2 \log \boldsymbol{f}}{\partial \theta_i \partial \theta_j}\right]. \tag{4.6}$$

(2) Fisher information matrix $I(\boldsymbol{\theta}) = E\left[\frac{\partial \log \boldsymbol{f}}{\partial \boldsymbol{\theta}}\cdot\left(\frac{\partial \log \boldsymbol{f}}{\partial \boldsymbol{\theta}}\right)^{\mathrm{T}}\right]$ satisfies finite and definite at $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}}$.

(3) There is an open subset Ω containing the real parameter $\bar{\boldsymbol{\theta}}$ in the parameter space ω, such that for all observation values $\boldsymbol{X}_i$, $(i = 1,2,\cdots,n)$, the density $\boldsymbol{f}$ admits all third-order derivatives

$$\frac{\partial^3 \log \boldsymbol{f}}{\partial \theta_i \partial \theta_j \partial \theta_k}, \quad \forall\, \boldsymbol{\theta} \in \omega, \tag{4.7}$$

and there exist function M_{ijk} such that

$$\left|\frac{\partial^3 \log \boldsymbol{f}}{\partial \theta_i \partial \theta_j \partial \theta_k}\right| \le M_{ijk}(\boldsymbol{X}), \quad \forall\, \boldsymbol{\theta} \in \omega, \tag{4.8}$$

where the element $m_{ijk} = E_{\bar{\boldsymbol{\theta}}}[M_{ijk}(\boldsymbol{X})] < \infty$ for i, j, k.

Theorem 4.3 *Assuming that all the regularization conditions* (1) − (3) *are satisfied. For given* $\lambda_1^* > 0$ *and* $\lambda_2^* > 0$, *if* $\frac{\lambda_1^{(n)}}{\sqrt{n}} \to \lambda_1^*$, $\frac{\lambda_2^{(n)}}{\sqrt{n}} \to \lambda_2^*$ *and* $\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{X}_i\boldsymbol{X}_i^T = \mathcal{C}$, *where* $\mathcal{C}$ *is non-singular. Then the estimation* $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_2)^T$ *under the CNet-RLR model satisfies the following oracle properties:*

(1) *Sparsity:* $P(\hat{\boldsymbol{\theta}}_2 = \mathbf{0}) \to 0$;

(2) *Asymptotic normality:* $\sqrt{n}(\hat{\boldsymbol{\theta}} - \bar{\boldsymbol{\theta}}) \xrightarrow{d} N(0, \mathcal{C}^{-1})$, *where* $\xrightarrow{d}$ *represents convergence in distribution.*

The proof of this theorem can be found in Appendix A.4. Proof of Theorem 4.3.

4.3 Convergence analysis

Theorem 4.4 ***(Barrier convergence theorem)*** *Let* $\Phi(\boldsymbol{\theta},\boldsymbol{u}) = -\mathcal{L}(\boldsymbol{\theta};\mathcal{D}) + \lambda_1\mathbf{1}^T\boldsymbol{u} + \frac{\lambda_2}{2}\boldsymbol{\theta}^T\mathcal{L}\boldsymbol{\theta}$ *be the objective function of the equivalent problem (3.1) with feasible solution domain*

$$\mathrm{dom}\,\Phi = \{(\boldsymbol{\theta},\boldsymbol{u}) \in \mathbb{R}^p \times \mathbb{R}^p : \boldsymbol{u} - \boldsymbol{\theta} \ge \mathbf{0}, \boldsymbol{u} + \boldsymbol{\theta} \ge \mathbf{0}\}.$$

Suppose that $\mathrm{dom}\,\Phi \neq \varnothing$, $\mathrm{dom}\,\phi_\mu \neq \varnothing$, *and the equivalent problem (3.1) has an optimal solution* $(\boldsymbol{\theta}^*, \boldsymbol{u}^*)$. *Then,* $\lim_{\mu\to 0^+}\phi_\mu(\boldsymbol{\theta},\boldsymbol{u}) = 0$, *and the limit of any convergence subsequence of* $\{(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu)\}$ *is an optimal solution to the equivalent problem* (3.1).

Theorem 4.5 ***(Algorithm convergence theorem)*** *Suppose that* $\{(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu, \boldsymbol{z}_\mu^\theta, \boldsymbol{z}_\mu^u)\}$ *is the result sequence obtained by Algorithm 2, then whose any limit point meets the first-order optimal conditions* (3.10).

These two theorems give qualitative results for the interior-point algorithm and provide the theoretical basis for the convergence of the algorithm. The proofs of them can be found in Appendix A.5. Proof of Theorem 4.4 and Appendix A.6. Proof of Theorem 4.5, respectively.

5 Experiments

CNet-RLR model predicts class labels by selecting a subset of features that are connected in the network. To evaluate its feature selection and prediction performance, we here conduct experiments both on a simulation dataset as well as a real-world dataset. For illustrating its effectiveness and efficiency, some representative models including four RLR models (i.e., Lasso-RLR [20], Enet-RLR [21], Net-RLR [30], AbNet-RLR [47]) and one graph embedded feature selection model (i.e., GEDFN [49]) are selected to compare with the performances of CNet-RLR method on these two datasets.

5.1 Experiment on simulation data

5.1.1 Synthetic dataset

We first generate a synthetic dataset $\mathcal{D} = \{\boldsymbol{X} \in \mathbb{R}^{n\times p},\ \boldsymbol{y} \in \{0, 1\}\}$. Specifically, it is created by the following model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\theta} + \boldsymbol{\varepsilon}, \tag{5.1}$$

with the coefficient vector and the error term

$$\boldsymbol{\theta} = [sample(N(0, 1), 40), \underbrace{0, 0, \cdots, 0}_{p-40}]^{\mathrm{T}}, \quad \boldsymbol{\varepsilon} \sim N(\boldsymbol{0}, \boldsymbol{\sigma}^2), \tag{5.2}$$

where $\boldsymbol{\theta}$ consists of two parts, one is the dense term $sample(N(0, 1), 40)$, which is a vector obtained by randomly sampling 40 times from the standard normal distribution, another is the sparse term, which is a zero vector with $p - 40$ elements.

Then, we generate an observation matrix $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ (e.g., gene expression profiles in microarray and RNA-seq data) and its corresponding binary response vector $\boldsymbol{y}$ (e.g., normal and disease phenotypes in cancer research) with the following formulas

$$\boldsymbol{X} \sim N(\boldsymbol{0}, \boldsymbol{\Sigma}), \quad \boldsymbol{y} \sim \text{Bernoulli}[\text{logit}^{-1}(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta})], \tag{5.3}$$

where $\boldsymbol{\Sigma} \in \mathbb{R}^{p\times p}$ with elements σ_{kj},

$$\sigma_{kj} = \begin{cases} 0.6, & 1 \le k \ne j \le 40, \\ 1, & k = j = 1, 2, \cdots, p, \\ 0, & \text{otherwise.} \end{cases}$$

It is noteworthy that these 40 features in matrix $\boldsymbol{X}$ are strong intercorrelated and they tend to be linked by edges in a priori network-structured features. Particularly, we propose a prior network expressed by $\boldsymbol{A} \in \mathbb{R}^{p\times p}$, whose first 40 vertexes are linked with a higher probability of 0.05 to form the network structure (sorted in the natural order of the nodes), and the rest are connected with a lower probability of 0.02. In the case of $n = 500$ and $p = 100$, we generate a synthetic dataset, Fig. 3a illustrates the feature network induced by $\boldsymbol{A}$.

5.1.2 Feature selection

Here, we perform the simulation studies on ten datasets generated by ten different random seeds to explore the performances of CNet-RLR method by comparing with Lasso-RLR, Enet-RLR, Net-RLR, AbNet-RLR and GEDFN methods. From each synthetic dataset, 300 samples (60%) are randomly selected as the training dataset, 200 rest samples (40%) are regarded as the testing dataset. To select the optimal parameters and compute their AUC values of binary classification in the testing data, all the RLR models are performed on the training dataset and 5-fold cross-validation (CV) is used to minimize their objective functions respectively. For GEDFN, all hyper-parameters associated with its classification performance are optimized by a Bayesian optimization algorithm [72] as that in its original paper [49]. GEDFN generates a ranking of features and then we select the top-ranked features as the selected features according to [49]. The comparison results of the number of features selected by the six models on the ten different training datasets and their AUC values on the corresponding testing datasets are shown in Fig. 4.

As shown in Fig. 4a, due to the dependence of the model on the data, there is no obvious rule for the number of features selected by different models on different datasets. For the AUC values shown in Fig. 4b, our proposed CNet-RLR model ranks the first on most of the datasets, except the AUC values on the 4-th and 7-th datasets which are slightly lower (less than 0.006) than those of AbNet-RLR model. By investigating the network structure of selected biomarkers, we found that the CNet-RLR model recognizes the features in the form a connected network, but AbNet-RLR model is not. Figure 4c intuitively shows the average AUC values of all models on the ten different datasets and their corresponding standard deviations. It clearly shows that our proposed model is better than other five models under many trials.

5.1.3 Comparison of features

To compare the features selected by CNet-RLR model with four different RLR models and GEDFN model, we generate the network structures with respect to these selected variables with non-zero coefficients for RLR models and with the top-ranked coefficients for GEDFN model, as well as the numbers of nodes and edges in the feature network. Taking the first synthetic dataset ($random.seed = 10$) as an example, the detailed feature selection results of these comparing methods on the testing dataset are shown in Table 1, where the percentages in parentheses indicate the

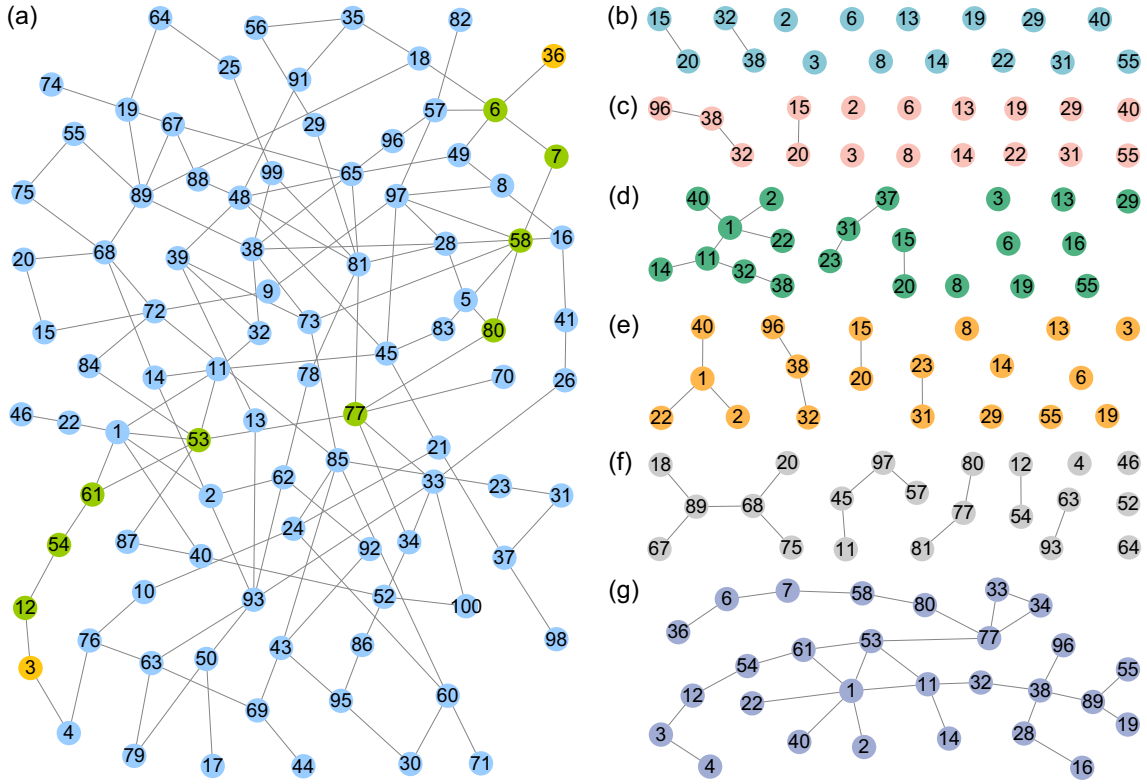

Fig. 3 The feature network and selected features, in which numbers on nodes only indicate the name for each node. **a** The simulated network of feature linkages. The features selected by **b** Lasso-RLR, **c** Enet-RLR, **d** Net-RLR, **e** AbNet-RLR, **f** GEDFN and **e** CNet-RLR

proportions of the features that have the connection edges in the prior linkage network in all the selected features by each method. It can be seen that CNet-RLR obtains a higher AUC value in the classification compared with the other methods. Also, it identifies much more edges in these features.

More specifically, Fig. 3b-f illustrates the subnetworks of features selected by Lasso-RLR, Enet-RLR, Net-RLR, AbNet-RLR, GEDFN and CNet-RLR, respectively. It is clear that Lasso-RLR and Enet-RLR only select out some isolated nodes with few linking edges in the feature network. Net-RLR, AbNet-RLR and GEDFN select relatively more nodes with linages in the given network than the former two methods. In fact, they do not consider the interrelationship between the selected features and they independently select the features disregarding the prior networked structure among them. The few edges as shown are extracted directly from the network. In contrast, CNet-RLR selects the features with the connectivity property when we impose the constraints on regression coefficients. The results indicate the CNet-RLR method can maintain the intrinsic structures in the original features and effectively select these relevant features performing better classification for distinguishing the two kinds of samples in the simulation data.

5.2 Experiment on real data

5.2.1 UCEC dataset

The feature selection in transcriptomics data is expected to discovery diagnostic biomarkers for UCEC [9]. We download the data of UCEC for training and independent testing (external verification) from TCGA hub of UCSC Xena [73] and NCBI GEO database (GSE63678), separately. Table 2 illustrates the data statistics, where the numbers in parentheses refer to the sample size.

5.2.2 Connected biomarker genes

A clear framework for detecting UCEC biomarkers from RNA-seq data by CNet-RLR model and validating them in gene expression data can be found in Appendix B. We select 360 differentially expressed genes (DEGs) among all genes and extract their prior network from RegNetwork [36]. As a result, the structured network containing 124 nodes/genes (accounting for 34.44%) with 177 edges (Fig. 5a) is obtained to be treated as the feature candidate set. We

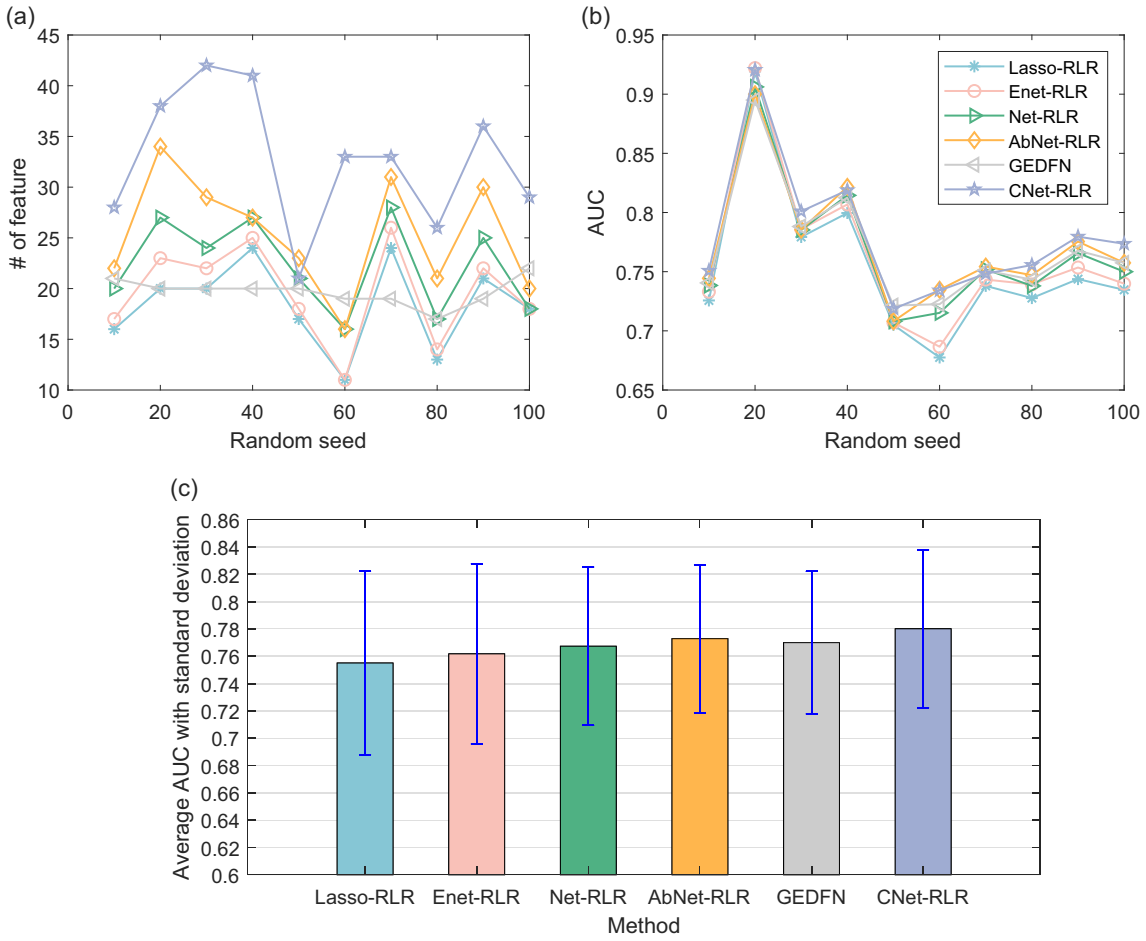

Fig. 4 The performance comparison of six models on ten different datasets, where each random seed generates a unique dataset. **a** The number of features. **b** The AUC. **c** The average AUC with standard deviation

Table 1 The comparisons of five RLR methods and GEDFN method in the simulation studies

Method	# of features	# of vertexes	# of edges	AUC
Lasso-RLR	16	4 (25.00%)	2	0.726
Enet-RLR	17	5 (29.41%)	3	0.733
Net-RLR	20	13 (65.00%)	10	0.738
AbNet-RLR	22	11 (50.00%)	7	0.744
GEDFN	21	17 (80.95%)	12	0.740
CNet-RLR	28	28 (100.00%)	30	0.751

Table 2 The detailed information of the two datasets used in the real data experiment

Dataset	# of samples	# of genes	Phenotype of sample	Reference
HiSeqV2	201	20,530	Normal (24)/UCEC (177)	[74]
GSE63678	12	13,749	Normal (5)/UCEC (7)	[75]

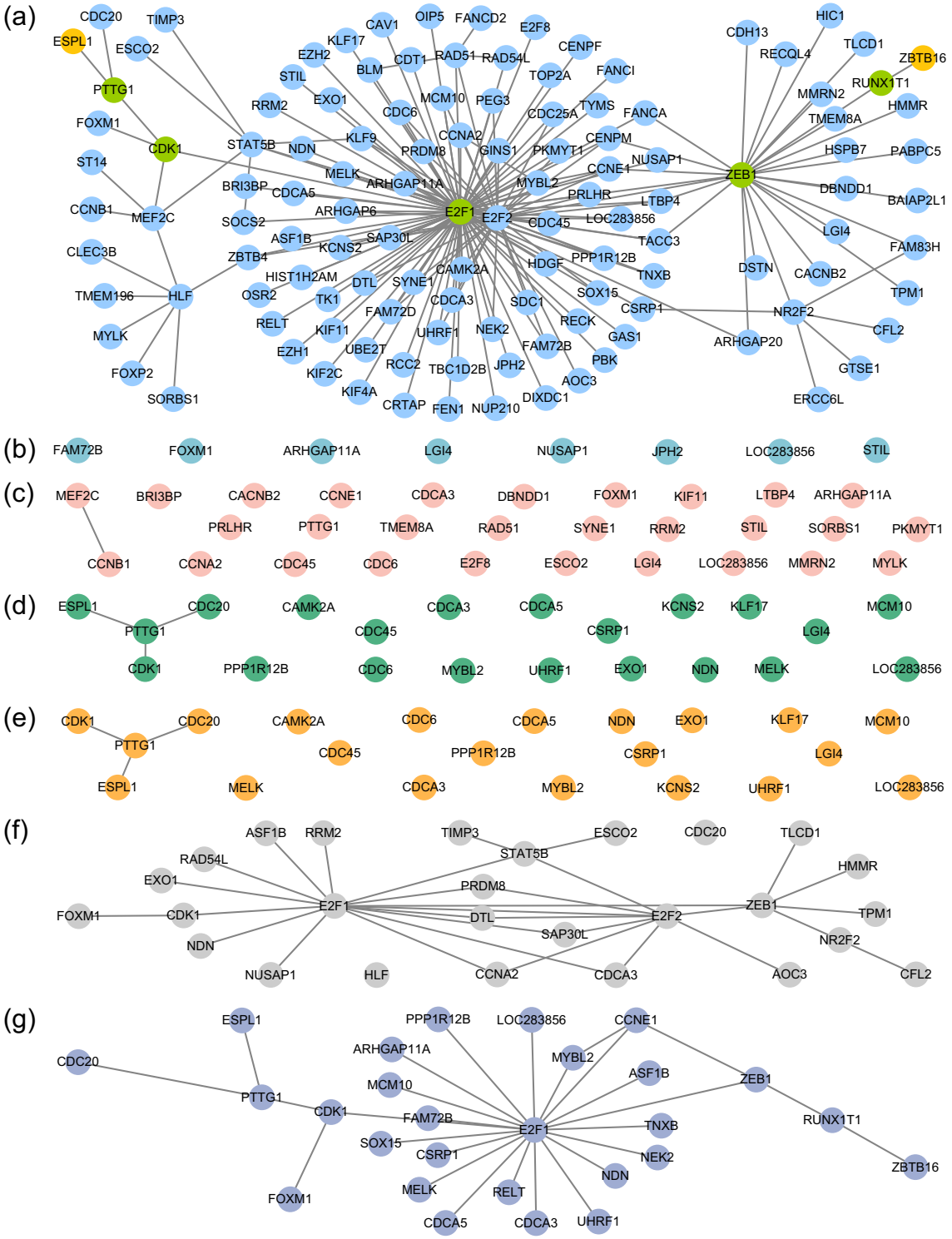

Fig. 5 The feature network and selected features. The words on nodes represent gene symbols. **a** The prior gene interaction network. The feature selected by **b** Lasso-RLR, **c** Enet-RLR, **d** Net-RLR, **e** AbNet-RLR, **f** GEDFN and **g** CNet-RLR

then perform feature selection and implement classification simultaneously in the training dataset, 75% samples of the new balanced dataset obtained by randomly sampling from the HiSeqV2 dataset, of UCEC using CNet-RLR. The selected features in this embedded classifier can be seen as UCEC biomarkers for discriminating different phenotypes of disease and normal samples. We chose the features whose corresponding coefficients $\theta_i \neq 0$, and finally identify 27

genes token as UCEC biomarkers. Figure 5g shows these biomarkers and their networking structure.

5.2.3 Comparison with other methods

To achieve a fair comparison, we implement the experiments in the same training dataset using these comparing methods according to the same steps. For each RLR model, 10-fold CV is applied to determine the optimal tuning parameter λ with minimal misclassification error in the training dataset, respectively. As for GEDFN, a Bayesian optimization algorithm is used to select the optimal hyperparameters via the classification performance [72]. Then we verify their classification performances in the testing dataset (the remaining 25% samples of the new balanced dataset), i.e., internal validation dataset.

Figure 5b-e shows that Lasso-RLR, Enet-RLR, Net-RLR and AbNet-RLR selects a certain number of feature genes, but few of them have a network structure and most of them exist in isolation. It is worthy to note that the features selected by AbNet-RLR are completely consistent with those selected by Net-RLR. Therefore, we will only analyze the features of Net-RLR in the subsequent section. For GEDFN, to be fair, we select the top-ranked 27 genes as the selected features to achieve the same number of features selected by CNet-RLR. Figure 5f shows that these features form three connected components, among which 25 genes have a connection relationship in the network, and the other two genes are isolated nodes. Compared with the others, Fig. 5g shows the feature genes selected by CNet-RLR. Clearly, we can see that these genes are not only connected to each other in the form of a network or a pathway, but also they own more functional implications that better explain the molecular mechanism of UCEC. The results obtained in the real UCEC data are consistent with those in the simulation data.

5.2.4 Functional enrichement analysis of biomarkers

For analyzing the functions implicated in these identified UCEC biomarkers, we perform a functional enrichment analysis of GO (gene ontology) [76]. The enriched GO terms of biomarkers selected by four RLR methods (Lasso-RLR, Enet-RLR, Net-RLR and CNet-RLR) and GEDFN are shown in Fig. S1 in the supplementary file, where each bar refers to a top-ranked non-redundant enriched GO cluster and its discrete color scale represents its statistical significance [77]. The enriched functions indicate most of these biomarkers are related to the regulation (positive/negative) of biological process (BP), cellular process, metabolic process, reproductive process and developmental process. It is worth mentioning that some of them have been reported in literature [78–82]. The enriched functions verify the interestingness of these identified biomarkers and also illustrate the effectiveness of CNet-RLR model (3.1) in feature selection at the same time.

In fact, it is an urgent demand to annotate the functions of biological networks with the help of the interactions between molecules. Fortunately, network ontology analysis (NOA) developed by [83] is such a method to find more relevant and specific functions for a specified network structure. Instead of considering the node-based enrichments, NOA implements an statistical test for each edge-based functional term. To further analyze the enriched functions underlying these selected biomarkers with edges, we apply NOA to enrich the functions in these biomarkers as shown in Fig. 5b-g. Some representative GO functions enriched by the edges of biomarkers are listed in Table 3, where $P.R.$ is the abbreviation of 'Positive regulation of'.

Table 3 illustrates some important functions, such as transcription, regulation of gene expression and RNA metabolic process, are all enriched significantly in the biomarkers discovered by CNet-RLR. These functions have been proved to be the hallmarks of the occurrence and development of cancer [84]. If we consider the identified network-based biomarkers, the enriched functional terms indicate they are meaningful of revealing cancer mechanism. Clearly, these functions have not been enriched in the biomarkers identified by other RLR methods. CNet-RLR provides more interpretable features in a connected subnetwork indicating crucial dysfunctions.

5.2.5 External validation

In order to further demonstrate the rationality of the identified UCEC biomarkers by CNet-RLR model, we verify them in the external validation dataset GSE63678. After statistical comparison, we found that there are 22 genes with expression values in independent datasets GSE63678 among the 27 detected biomarkers. To begin with, we train the LR classifier (2.3) using the expression value of 22 feature genes (denoted by $x_{i1}, x_{i2}, \cdots, x_{i,22}$) in the training dataset to obtain each regression coefficient and the intercept. Next, we use the parameters from the trained classifier in the previous step and combine the corresponding expression values of 22 genes in the external dataset to predict the response variable $\boldsymbol{y}$.

Particularly, Table 4 shows the number of features selected by each method in the discovery dataset, the number of edges linked by these features, and the five classification evaluation indicators in the external validation dataset. It is found that CNet-RLR obtains a better classification performance than the other methods in the external testing dataset, which reflects the effectiveness of CNet-RLR model in identifying UCEC biomarkers by feature selection.

Table 3 The functions analysis results of four RLR models and GEDFN model by NOA

Method	GO ID	*P*.Val	Annotated edges	GO term
Lasso-RLR	– –	– –	– –	– –
Enet-RLR	GO:0006351	1.000	{MEF2C-CCNB1}	Transcription DNA-templated
	GO:0006464	1.000		Cellular protein modification process
	GO:0007165	1.000		Signal transduction
Net-RLR	GO:0061982	1.000	{ESPL1-PTTG1} $\triangleq \mathcal{E}_{Net1}$	Meiosis I cell cycle process
	GO:0051784	1.000	{PTTG1-CDC20} $\triangleq \mathcal{E}_{Net2}$	Negative regulation of nuclear division
	GO:0006281	1.000	{PTTG1-CDK1} $\triangleq \mathcal{E}_{Net3}$	DNA repair
	GO:0051321	1.000	$\mathcal{E}_{Net1} \bigcup \mathcal{E}_{Net2}$	Meiotic cell cycle
	GO:0009056	1.000	$\mathcal{E}_{Net2} \bigcup \mathcal{E}_{Net3}$	Catabolic process
	GO:0000003	1.000	$\mathcal{E}_{Net1} \bigcup \mathcal{E}_{Net2} \bigcup \mathcal{E}_{Net3}$	Reproduction
GEDFN	GO:0009893	0.009	$\mathcal{E}_{GEDFN2} \bigcup$ {DTL-E2F1, DTL-E2F2}	*P.R.* metabolic process
	GO:0010604	0.009		*P.R.* macromolecule metabolic process
	GO:0051173	0.009		*P.R.* nitrogen compound metabolic process
	GO:2000026	0.020	$\mathcal{E}_{GEDFN1}$[1]	Regulation of multicellular organismal development
	GO:0009891	0.028	$\mathcal{E}_{GEDFN2}$[2]	*P.R.* biosynthetic process
	GO:0010557	0.028		*P.R.* macromolecule biosynthetic process
	GO:0010628	0.028		*P.R.* gene expression
	GO:0031325	0.028		*P.R.* cellular metabolic process
	GO:0031328	0.028		*P.R.* cellular biosynthetic process
CNet-RLR	GO:0006351	0.036	$\mathcal{E}_{CNet1}$[3]	Transcription
	GO:0010467	0.036		Gene expression
	GO:0010468	0.036		Regulation of gene expression
	GO:0016070	0.036		RNA metabolic process
	GO:0097659	0.036		Nucleic acid-templated transcription
	GO:0009889	0.052	$\mathcal{E}_{CNet1} \bigcup$ {NEK2-E2F1} $\triangleq \mathcal{E}_{CNet2}$	Regulation of biosynthetic process
	GO:0019438	0.052		Aromatic compound biosynthetic process
	GO:0031326	0.052		Regulation of cellular biosynthetic process
	GO:1901362	0.052		Organic cyclic compound biosynthetic process
	GO:0009058	0.076	$\mathcal{E}_{CNet2} \bigcup$ {MCM10-E2F1}	Biosynthetic process
	GO:0009059	0.076		Macromolecule biosynthetic process
	GO:0044249	0.076		Cellular biosynthetic process
	GO:1901576	0.076		Organic substance biosynthetic process

[1] $\mathcal{E}_{GEDFN1} \triangleq$ {E2F1-CDK1, E2F2-E2F1, STAT5B-E2F1, ZEB1-E2F1, STAT5B-E2F2, ZEB1-E2F2}

[2] $\mathcal{E}_{GEDFN2} \triangleq$ {E2F1-CCNA2, E2F2-CCNA2, E2F1-CDK1, FOXM1-CDK1, E2F2-E2F1, NDN-E2F1, STAT5B-E2F1, ZEB1-E2F1, STAT5B-E2F2, ZEB1-E2F2, NR2F2-ZEB1}

[3] $\mathcal{E}_{CNet1} \triangleq$ {ZEB1-RUNX1T1, ZBTB16-RUNX1T1, E2F1-CCNE1, MYBL2-CCNE1, ZEB1-CCNE1, E2F1-CDK1, FOXM1-CDK1, MYBL2-E2F1, NDN-E2F1, SOX15-E2F1, ZEB-E2F1, UHRF1-E2F1}

Table 4 The performances of five RLR models and GEDFN model on the UCEC dataset

Method	Training (HiSeqV2)		Validation (GSE63678)						
	# of features	# of edges	# of overlaps	# of edges	Acc	Pre	Sn	Sp	F- measure
Lasso-RLR	8	0	5	0	0.833	0.800	0.800	0.857	0.800
Enet-RLR	29	1	24	1	0.750	0.750	0.600	0.857	0.667
Net-RLR	21	3	15	3	0.833	1.000	0.600	1.000	0.750
AbNet-RLR	21	3	15	3	0.833	1.000	0.600	1.000	0.750
GEDFN	27	31	24	28	0.833	0.800	0.800	0.857	0.800
CNet-RLR	27	28	22	23	0.917	1.000	0.800	1.000	0.889

*Acc: accuracy, Pre: precision, Sn: sensitivity, Sp: specificity

More clearly, Fig. 6 shows the ROC curves in the classifications with the biomarkers selected by six feature selection methods. CNet-RLR achieves a higher AUC value of 0.900 when compared to the lower values obtained by Lasso-RLR, Enet-RLR, Net-RLR, AbNet-RLR and GEDFN.

6 Discussion

Traditional classification models with embedded feature selection are generally developed based on the assumption that each individual feature has an individual contribution to the sample category. The inherent network structure of these features are ignored in the models. Therefore, when they are tested in different datasets, the predictive performances are often unexpectedly low, even if these validation datasets are obtained from similar genomic scenarios [35]. This is because genes often work together, and the disease-related genes are always involved in the same pathway. Moreover, the category of a sample depends not only on some independent genes, but also on its neighboring genes in the form of a network or a pathway [35].

The CNet-RLR model solves this problem by introducing the constraints of feature interaction in a natural way. In the model, we formulate it as a constrained mathematical programming problem. The connectivity constraints ensure that the selected molecular biomarkers are embedded with the network connectivity, while the other existing models that do not consider connectivity and only get

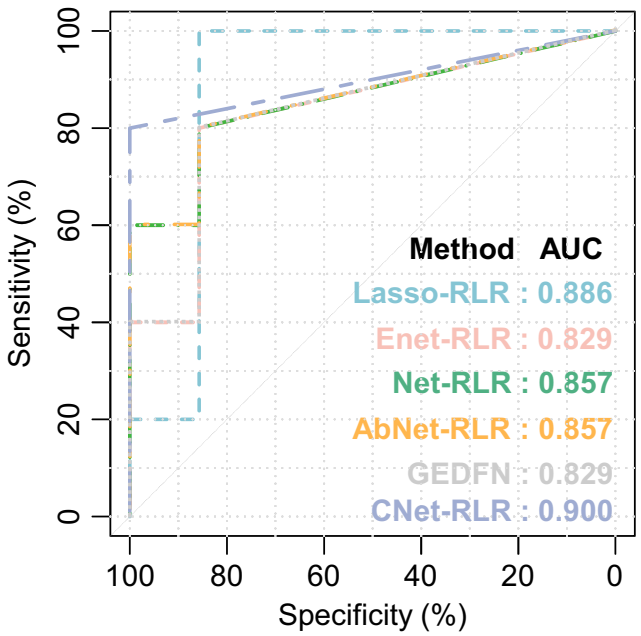

Fig. 6 The ROC curves of five RLR models and GEDFN model in the external validation dataset

some isolated points or points that coexist in small subnetworks. The experiments have illustrated the CNet-RLR model directly recognizes the biomolecular network and effectively interprets the disease mechanism while still maintaining important classification attributes.

In particular, we here discuss in depth the consistency of the candidate gene network established in Section 5.2.2 with algebraic connectivity. Figure 5a shows the network with 124 nodes and 177 edges, which is composed of candidate genes after dimensionality reduction. From the connection architecture of the 124 genes in the form of a network, we get its adjacency matrix A and normalized Laplacian matrix $\mathcal{L}$. The eigenvalues of $\mathcal{L}$ and its corresponding eigenvectors can be obtained. Sorting the 124 eigenvalues in an ascending order, i.e., $-2.511e-15 = \lambda_1 \leq \lambda_2 \ (= 0.074) \leq \cdots \leq \lambda_{123} \ (= 1.925) \leq \lambda_{124} = 1.931$, we find that the smallest eigenvalue $\lambda_1 \approx 0$ and the largest eigenvalue $\lambda_{124} \approx 2$. Indeed, the second smallest eigenvalue $\lambda_2 \neq 0$. As what Lemma 2.1 promised, it is straightforward to show that the features we selected satisfy the algebraic connectivity conditions. All eigenvalues and eigenvectors of this matrix are shown in Table S1 of the supplementary file.

More specifically, we also check the relationship and difference between the biomarkers identified by different models from the perspectives of feature selection and functional analysis mentioned in Section 5.2.4. Figure 7a shows the details of the overlapping biomarker genes selected by four RLR models and GEDFN model. The feature genes selected by each model are listed in the supplementary Table S2. From Fig. 7a, we find that among the 27 feature genes selected by CNet-RLR, 4 genes are identified by Lasso-RLR, 6 genes are identified by Enet-RLR, 14 genes are identified by Net-RLR (also AbNet-RLR), 8 genes are identified by GEDFN. Besides, for the GEDFN method, 14 genes are independently selected by its own. In particular, it should be noted that the features selected by GEDFN are different in each training step and it is not a very interpretable model in classification and feature selection. Compared with the other RLR alternatives and the deep learning method, it is exciting that CNet-RLR independently discovers 6 novel feature genes.

In these node-based enriched functions, after a simple statistical analysis to those enriched GO terms, the overlapping status of enriched functions in these biomarkers identified by four RLR models is shown in Fig. 7b. Among the enriched functions in the biomarkers selected by CNet-RLR, 7 of them are also enriched by the other RLR methods. It indicates the reliability of our proposed CNet-RLR method in feature selection from the perspective of functional implications. As shown in Fig. 7, the gene architecture selected by CNet-RLR with a priori information of feature interaction and connectivity constraint is very different from those selected by the other RLR models. We need further to check the functional implications of these interacting edges in the connected feature network.

For these edge-based enriched functions as shown in Table 3. Inevitably, none of edge between the biomarker genes has been picked out by Lasso-RLR, so there is no such result. In contrast, the biomarker genes selected by Enet-RLR contain only one edge, so it implies less functions. Similarly, the biomarkers selected by Net-RLR have three edges, and few functions are detected. However, they are not statistical significant (P.val $= 1.000$). The biomarker genes selected by GEDFN enrich significant functions. The number of enriched GO terms (total 406) contains only one less than those of CNet-RLR. However, the network-embedded feature selection is based on deep feedforward neural networks and the selected features are not stable when it chooses different hyper-parameters. On the contrary, CNet-RLR does not contain such parameter conversion process, but makes the identified biomarkers directly form a network, which well explains that the collaborative work between genes is closely related to the

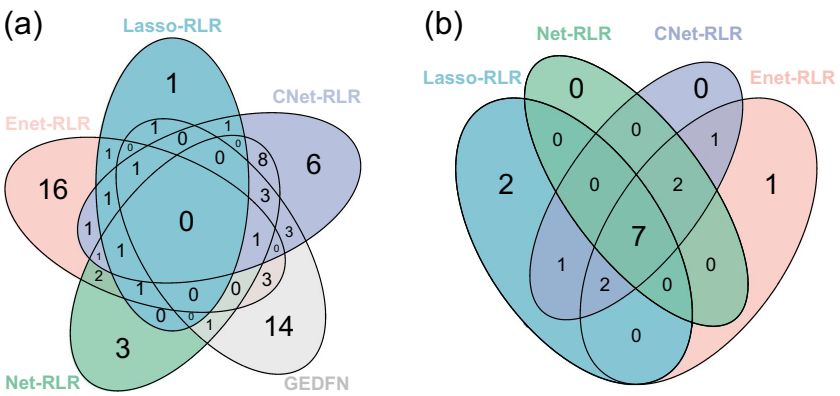

Fig. 7 Overlaps of the comparing of different models. **a** The biomarkers. **b** The enriched functions

dysfunctions of UCEC. This provides us with new idea for understanding the potential mechanism of the progress and transfer of UCEC [85].

In the existing RLR methods, they have not recognized the exact interacting structure within features. However, our proposed CNet-RLR model embeds the connectivity constraints in the network-based feature selection procedure. From the results on both simulation and real datasets, the CNet-RLR model can incorporate network structure in feature selection effectively and efficiently. In the comparison study, the experimental results prove CNet-RLR achieves higher prediction accuracy and better generalization performance. When applying in the high-throughput transcriptomic data, it can not only discover potential diagnostic feature genes, but also dig out the interrelationship and internal structure between these genes that indicate a pathway of performing critical dysfunctions.

7 Conclusion

In this study, we proposed a novel embedded feature selection method for regularized logistic regression, called CNet-RLR. It takes the connectivity constraints between variables into account. CNet-RLR is formulated as a constrained optimization problem and solved by designing an interior-point algorithm. Theoretically, we proved its grouping effect and oracle property. The feasibility and effectiveness of the interior-point algorithm are guaranteed by the convergence analysis. In the aspect of feature selection, in order to illustrate the superiority and effectiveness of CNet-RLR model, we implemented two experiments on both the simulation dataset and the real UCEC dataset. Moreover, compared with other RLR alternatives and other network-embedded strategy, CNet-RLR incorporates the networking structure underlying features by introducing the connectivity constraints. It outperforms these alternative methods in accuracy and efficiency. For recognizing the network-structured between variables, the features selected by CNet-RLR contain more sense of the data. It is easy to interpret the meaning and mechanism related to these selected features. Thus, it benefits a better understanding of our findings. Obviously, CNet-RLR exampled in biomarker discovery from genomics data can be easily extended for other kinds of multidimensional data.

Acknowledgements The authors would like to thank the editor and anonymous reviewers for their valuable comments and suggestions which greatly improve our paper. We also thank the members in our lab for their assistance in the project.

Author Contributions LL proposed the model and conducted the experiments, analysed the data and drafted the manuscript. ZL proposed the idea and supervised the studies, coordinated the project and revised the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding This work was partially supported by the National Key Research and Development Program of China under grant number 2020YFA0712402; National Natural Science Foundation of China (NSFC) under grant numbers 61973190 and 61572287; Shandong Provincial Key Research and Development Program (Major Scientific and Technological Innovation Project) under grant number 2019JZZY010423; the Innovation Method Fund of China (Ministry of Science and Technology of China) under grant number 2018IM020200; the Program of Qilu Young Scholars of Shandong University.

Availability of supplementary files and source codes All supplementary files and source codes used in this paper can be available at https://github.com/zpliulab/CNet.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

Appendix A

In this appendix, we provide the proofs of theorems.

A.1. Proof of Theorem 2.1

In order to prove Theorem 2.1, we firstly give the following preliminary lemma.

Lemma A.1 [41] *Let $\mathcal{L} \in \mathbb{R}^{n\times n}$ be an real symmetric matrix, with n eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$. Then $\mathcal{L}$ has n mutually orthogonal unit eigenvectors $\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2, \cdots, \boldsymbol{\epsilon}_n$, with*

$$\mathcal{L}\boldsymbol{\epsilon}_i = \lambda_i \boldsymbol{\epsilon}_i \quad (i = 1, 2, \cdots, n) \quad \text{and} \quad Q^T \mathcal{L} Q = \Lambda, \tag{A.1}$$

where Q is the unitary matrix with columns $\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2, \cdots, \boldsymbol{\epsilon}_n$ and $\Lambda = diag(\lambda_1, \lambda_2, \cdots, \lambda_n)$.

Proof As a $n \times n$ real symmetric matrix, $\mathcal{L}$ has n linearly independent eigenvectors corresponding to the eigenvalue $\lambda_1, \lambda_2, \cdots, \lambda_n$, denoted as $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \cdots, \boldsymbol{\alpha}_n$.

At first, we orthogonalize $\boldsymbol{\alpha}_i$ $(i = 1, 2, \cdots, n)$. Let

$$\boldsymbol{\beta}_i = \begin{cases} \boldsymbol{\alpha}_i, & i = 1, \\ \boldsymbol{\alpha}_i - \sum_{j=1}^{i-1} \frac{\langle \boldsymbol{\alpha}_i, \boldsymbol{\beta}_j \rangle}{\langle \boldsymbol{\beta}_j, \boldsymbol{\beta}_j \rangle} \boldsymbol{\beta}_j, & i = 2, 3, \cdots, n, \end{cases}$$

we have $\langle \boldsymbol{\beta}_i, \boldsymbol{\beta}_j \rangle = 0$ when $i \neq j$ where $i, j = 1, 2, \cdots, n$.

Then we proceed to unitize $\boldsymbol{\beta}_i (i = 1, 2, \cdots, n)$. Let

$$\boldsymbol{\epsilon}_i = \frac{\boldsymbol{\beta}_i}{\|\boldsymbol{\beta}_i\|_2},$$

we obtain

$$\langle \boldsymbol{\epsilon}_i, \boldsymbol{\epsilon}_j \rangle = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

It shows that $\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2, \cdots, \boldsymbol{\epsilon}_n$ is a set of standard orthogonal vectors of matrix $\mathcal{L}$.

In addition, according to Lemma A.1, we can know that there is an orthogonal matrix $Q \in \mathbb{R}^{n\times n}$ such that

$$\mathcal{L} = Q\Lambda Q^{\mathrm{T}}, \tag{A.2}$$

where $\Lambda = diag(\lambda_1, \lambda_2, \cdots, \lambda_n)$. Here we only need to make $Q = [\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2, \cdots, \boldsymbol{\epsilon}_n]$ to satisfy (A.2). Therefore, for any $\boldsymbol{v} \in \mathbb{R}^n$, it drives

$$\boldsymbol{v}^{\mathrm{T}}\mathcal{L}\boldsymbol{v} = \boldsymbol{v}^{\mathrm{T}}(Q\Lambda Q^{\mathrm{T}})\boldsymbol{v} = (Q^{\mathrm{T}}\boldsymbol{v})^{\mathrm{T}}\Lambda(Q^{\mathrm{T}}\boldsymbol{v}). \tag{A.3}$$

Let $\boldsymbol{p} = Q^{\mathrm{T}}\boldsymbol{v} \in \mathbb{R}^n$, suppose $\boldsymbol{p} = (p_1, p_2, \cdots, p_n)^{\mathrm{T}}$, we get

$$p_i = \langle \boldsymbol{\epsilon}_i, \boldsymbol{v} \rangle, \tag{A.4}$$

where $\langle\cdot\rangle$ represents the product function with $\langle \boldsymbol{\epsilon}_i, \boldsymbol{v}\rangle = \boldsymbol{\epsilon}_i^{\mathrm{T}}\boldsymbol{v}$. Thus we have

$$\boldsymbol{v}^{\mathrm{T}}\mathcal{L}\boldsymbol{v} = \boldsymbol{p}^{\mathrm{T}}\Lambda\boldsymbol{p} = \sum_{i=1}^{n}\lambda_i p_i^2 = \sum_{i=1}^{n}\lambda_i\langle \boldsymbol{\epsilon}_i, \boldsymbol{v}\rangle^2. \tag{A.5}$$

Now, we only consider the non-zero vector $\boldsymbol{v}$ orthogonal to $\boldsymbol{\alpha}_1$, it is clearly that $\boldsymbol{v}$ is orthogonal to $\boldsymbol{\epsilon}_1$. Combining with (A.4), we get

$$\sum_{i=1}^{n}\lambda_i\langle \boldsymbol{\epsilon}_i, \boldsymbol{v}\rangle^2 = \sum_{i=2}^{n}\lambda_i\langle \boldsymbol{\epsilon}_i, \boldsymbol{v}\rangle^2 \geq \lambda_2\sum_{i=2}^{n}\langle \boldsymbol{\epsilon}_i, \boldsymbol{v}\rangle^2 = \lambda_2\sum_{i=1}^{n}\langle \boldsymbol{\epsilon}_i, \boldsymbol{v}\rangle^2. \tag{A.6}$$

However,

$$\langle \boldsymbol{\epsilon}_i, \boldsymbol{v}\rangle^2 = (\boldsymbol{\epsilon}_i^{\mathrm{T}}\boldsymbol{v})^{\mathrm{T}}(\boldsymbol{\epsilon}_i^{\mathrm{T}}\boldsymbol{v}) = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{\epsilon}_i\boldsymbol{\epsilon}_i^{\mathrm{T}}\boldsymbol{v} = \boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}, \tag{A.7}$$

So taking (A.5) - (A.7) into consideration, we have

$$\boldsymbol{v}^{\mathrm{T}}\mathcal{L}\boldsymbol{v} \geq \lambda_2\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}. \tag{A.8}$$

The equal sign holds iff $\boldsymbol{v} = \boldsymbol{\alpha}_2$. As a consequence, for the second smallest eigenvalue λ_2, we get

$$\lambda_2 = \min_{\substack{\boldsymbol{v}\neq\boldsymbol{0} \\ \boldsymbol{v}\perp\boldsymbol{\alpha}_1}} \left\{ \frac{\boldsymbol{v}^{\mathrm{T}}\mathcal{L}\boldsymbol{v}}{\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}} \right\}. \tag{A.9}$$

Finally, we only need to make $\boldsymbol{\alpha}_1 = \boldsymbol{u}$, then Theorem 2.1 is proved. □

A.2. Proof of Theorem 4.1

Proof The proof is by contradiction. For any given non-negative regularization parameters λ_1 and λ_2, let

$$\mathcal{J}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\lambda}) = -\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) + \lambda_1\sum_{j=1}^{p}|\theta_j| + \frac{\lambda_2}{2}\boldsymbol{\theta}^{\mathrm{T}}\mathcal{L}\boldsymbol{\theta}. \tag{A.10}$$

If $\hat{\theta}_k \neq \hat{\theta}_j$, construct a new variable $\hat{\boldsymbol{\theta}}^*$ with elements $\hat{\theta}_l^*$ such that

$$\hat{\theta}_l^* = \begin{cases} \hat{\theta}_l, & \text{if } l \neq k \text{ and } l \neq j, \\ \dfrac{\hat{\theta}_k + \hat{\theta}_j}{2}, & \text{if } l = k \text{ or } l = j. \end{cases} \tag{A.11}$$

Since $\hat{\boldsymbol{X}}_k = \hat{\boldsymbol{X}}_j$, it is obvious that $\boldsymbol{X}\hat{\boldsymbol{\theta}}^* = \boldsymbol{X}\hat{\boldsymbol{\theta}}$. So as for $1 \leq i \leq n$, we have

$$\boldsymbol{X}_i^{\mathrm{T}}\hat{\boldsymbol{\theta}}^* = \boldsymbol{X}_i^{\mathrm{T}}\hat{\boldsymbol{\theta}}. \tag{A.12}$$

For the first term on the right side of (A.10), after simple calculations, (2.3) is equivalent to

$$-\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) = -\sum_{i=1}^{n}\left\{ y_i X_i^{\mathrm{T}}\boldsymbol{\theta} - \log\left(1+\exp(X_i^{\mathrm{T}}\boldsymbol{\theta})\right)\right\}, \tag{A.13}$$

Combining with (A.12), we get $-\mathcal{L}(\hat{\boldsymbol{\theta}}^*; \mathcal{D}) = -\mathcal{L}(\hat{\boldsymbol{\theta}}; \mathcal{D})$.

However, for the penalty $\lambda_1\sum_{j=1}^{p}|\theta_j| + \frac{\lambda_2}{2}\boldsymbol{\theta}^{\mathrm{T}}\mathcal{L}\boldsymbol{\theta}$, it is strictly convex, so we have

$$\lambda_1\sum_{j=1}^{p}|\hat{\theta}_j^*| + \frac{\lambda_2}{2}\hat{\boldsymbol{\theta}}^{*\mathrm{T}}\mathcal{L}\hat{\boldsymbol{\theta}}^* < \lambda_1\sum_{j=1}^{p}|\hat{\theta}_j| + \frac{\lambda_2}{2}\hat{\boldsymbol{\theta}}^{\mathrm{T}}\mathcal{L}\hat{\boldsymbol{\theta}}. \tag{A.14}$$

Substituting (A.13) and (A.14) into (A.10), it derives

$$\mathcal{J}(\hat{\boldsymbol{\theta}}^*; \mathcal{D}, \boldsymbol{\lambda}) < \mathcal{J}(\hat{\boldsymbol{\theta}}; \mathcal{D}, \boldsymbol{\lambda}). \tag{A.15}$$

Equation (A.15) shows that $\hat{\boldsymbol{\theta}}$ cannot be the minimizer of (A.10), which lead to a contradiction to (4.1). Consequently, we obtain $\hat{\theta}_k = \hat{\theta}_j$. □

A.3. Proof of Theorem 4.2

Proof Since $\hat{\theta}_k(\lambda_1, \lambda_2)\hat{\theta}_j(\lambda_1, \lambda_2) > 0$, $e_{kj} = 1$, and $d_k = d_j = w_{kj}$, so we have

$$sign(\hat{\theta}_k(\lambda_1, \lambda_2)) = sign(\hat{\theta}_j(\lambda_1, \lambda_2)). \tag{A.16}$$

According to (4.1) and (A.10), then $\hat{\boldsymbol{\theta}}(\lambda_1, \lambda_2)$ satisfies

$$\frac{\partial\mathcal{J}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\lambda})}{\partial\theta_i}\Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}(\lambda_1,\lambda_2)} = 0, \quad \text{if } \hat{\theta}_i(\lambda_1, \lambda_2) \neq 0. \tag{A.17}$$

Combining with (A.13), we have

$$\begin{aligned} -\nabla_\theta\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) &= -\sum_{i=1}^{n}\left\{ y_i\boldsymbol{X}_i^{\mathrm{T}} - \frac{\exp(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta})\cdot\boldsymbol{X}_i^{\mathrm{T}}}{1+\exp(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta})}\right\} \\ &= -\sum_{i=1}^{n}\left\{ y_i\boldsymbol{X}_i^{\mathrm{T}} - f\left(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta}\right)\cdot\boldsymbol{X}_i^{\mathrm{T}}\right\} \\ &= -\sum_{i=1}^{n}\boldsymbol{X}_i\left\{ y_i - f\left(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta}\right)\right\}, \\ &= \boldsymbol{X}^{\mathrm{T}}(\boldsymbol{f} - \boldsymbol{y}), \end{aligned} \tag{A.18}$$

where

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n, \ \boldsymbol{f} = \begin{bmatrix} f(\boldsymbol{X}_1^{\mathrm{T}}\boldsymbol{\theta}) \\ f(\boldsymbol{X}_2^{\mathrm{T}}\boldsymbol{\theta}) \\ \vdots \\ f(\boldsymbol{X}_n^{\mathrm{T}}\boldsymbol{\theta}) \end{bmatrix} \in \mathbb{R}^n$$

are the n-dimensional column vectors.

Thus (A.17) is equivalent to the following equation

$$(\hat{\boldsymbol{f}} - \boldsymbol{y})^{\mathrm{T}}\hat{\boldsymbol{X}}_i + \lambda_1 sign(\hat{\theta}_i) + \lambda_2 \hat{\boldsymbol{\theta}}^{\mathrm{T}}\mathcal{L}_i = 0, \tag{A.19}$$

where $\hat{\boldsymbol{X}}_i = (x_{1i}, x_{2i}, \cdots, x_{ni})^{\mathrm{T}} \in \mathbb{R}^n$ is the i-th column of the matrix $\boldsymbol{X} \in$, $\hat{\boldsymbol{f}} = (f(\boldsymbol{X}_1^{\mathrm{T}}\hat{\boldsymbol{\theta}}), f(\boldsymbol{X}_2^{\mathrm{T}}\hat{\boldsymbol{\theta}}), \cdots, f(\boldsymbol{X}_n^{\mathrm{T}}\hat{\boldsymbol{\theta}}))^{\mathrm{T}}$, and $\mathcal{L}_i$ is the i-th column of the normalized Laplacian matrix $\mathcal{L}$.

So, when $i = k$ and $i = j$, we have

$$(\hat{\boldsymbol{f}} - \boldsymbol{y})^{\mathrm{T}}\hat{\boldsymbol{X}}_k + \lambda_1 sign(\hat{\theta}_k) + \lambda_2 \hat{\boldsymbol{\theta}}^{\mathrm{T}}\mathcal{L}_k = 0, \tag{A.20}$$

$$(\hat{\boldsymbol{f}} - \boldsymbol{y})^{\mathrm{T}}\hat{\boldsymbol{X}}_j + \lambda_1 sign(\hat{\theta}_j) + \lambda_2 \hat{\boldsymbol{\theta}}^{\mathrm{T}}\mathcal{L}_j = 0. \tag{A.21}$$

Subtracting (A.21) from (A.20), and considering (A.16), we get

$$(\hat{\boldsymbol{f}} - \boldsymbol{y})^{\mathrm{T}}(\hat{\boldsymbol{X}}_k - \hat{\boldsymbol{X}}_j) + \lambda_2 \hat{\boldsymbol{\theta}}^{\mathrm{T}}(\mathcal{L}_k - \mathcal{L}_j) = 0. \tag{A.22}$$

According to the definition of $\mathcal{L}$ and the condition $d_k = d_j = w_{kj}$, we obtain

$$\hat{\boldsymbol{\theta}}^{\mathrm{T}}(\mathcal{L}_k - \mathcal{L}_j) = \frac{w_{kj}}{\sqrt{d_k d_j}}\hat{\theta}_k - \frac{w_{jk}}{\sqrt{d_j d_k}}\hat{\theta}_j = \hat{\theta}_k - \hat{\theta}_j. \tag{A.23}$$

Submitting (A.23) into (A.22), we have

$$\hat{\theta}_k - \hat{\theta}_j = \frac{(\boldsymbol{y} - \hat{\boldsymbol{f}})^{\mathrm{T}}(\hat{\boldsymbol{X}}_k - \hat{\boldsymbol{X}}_j)}{\lambda_2}. \tag{A.24}$$

Using the Cauchy-Schwartz inequality and the definition of the L_2-norm, we get

$$\hat{\theta}_k - \hat{\theta}_j \leq \frac{\sqrt{\sum_{i=1}^n (\boldsymbol{y} - \hat{\boldsymbol{f}})_i^2}\sqrt{(\hat{\boldsymbol{X}}_k - \hat{\boldsymbol{X}}_j)_i^2}}{\lambda_2} = \frac{\|\boldsymbol{y} - \hat{\boldsymbol{f}}\|_2 \cdot \|\hat{\boldsymbol{X}}_k - \hat{\boldsymbol{X}}_j\|_2}{\lambda_2}. \tag{A.25}$$

Taking the absolution to both sides of (A.25), we obtain

$$|\hat{\theta}_k - \hat{\theta}_j| \leq \frac{\|\boldsymbol{y} - \hat{\boldsymbol{f}}\|_2 \cdot \|\hat{\boldsymbol{X}}_k - \hat{\boldsymbol{X}}_j\|_2}{\lambda_2}. \tag{A.26}$$

According to the equivalence of L_2-norm and L_1-norm, it holds

$$\|\boldsymbol{y} - \hat{\boldsymbol{f}}\|_2 \cdot \|\hat{\boldsymbol{X}}_k - \hat{\boldsymbol{X}}_j\|_2 \leq c\, \|\boldsymbol{y} - \hat{\boldsymbol{f}}\|_1 \cdot \|\hat{\boldsymbol{X}}_k - \hat{\boldsymbol{X}}_j\|_1, \tag{A.27}$$

where c is a constant.

Since $\hat{\boldsymbol{\theta}}$ is the minimizer of Problem (2.18), the residual satisfies

$$\begin{aligned} \|\boldsymbol{y} - \hat{\boldsymbol{f}}\|_1 < \|\boldsymbol{y} - \boldsymbol{f}\|_1 &= \sum_{i=1}^n |y_i - f(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta})| \\ &\leq \sum_{i=1}^n \Big\{ |y_i| + |f(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta})| \Big\} = \|\boldsymbol{y}\|_1 + \|\boldsymbol{f}\|_1. \end{aligned} \tag{A.28}$$

While for the second term in right side of the equal sign of (A.28), we have

$$\lim_{\boldsymbol{\theta}\to\infty} \|\boldsymbol{f}\|_1 = \lim_{\boldsymbol{\theta}\to\infty} \left\| \frac{\exp \boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta}}{1 + \exp \boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta}} \right\|_1 = 0. \tag{A.29}$$

Taking the limit of $\boldsymbol{\theta} \to \infty$ to both sides of (A.26), and combining with (A.26)–(A.29), we have

$$|\hat{\theta}_k - \hat{\theta}_j| \leq \frac{c \cdot \|\boldsymbol{y}\|_1 \cdot \|\hat{\boldsymbol{X}}_k - \hat{\boldsymbol{X}}_j\|_1}{\lambda_2}. \tag{A.30}$$

Since $\boldsymbol{X}$ is standardized, then

$$\|\hat{\boldsymbol{X}}_k - \hat{\boldsymbol{X}}_j\|_1 = \sqrt{2(1-\rho)}, \tag{A.31}$$

where $\rho = \hat{\boldsymbol{X}}_k^{\mathrm{T}}\hat{\boldsymbol{X}}_j$.

Combining with (A.30) and (A.31), it holds

$$\frac{|\hat{\theta}_k - \hat{\theta}_j|}{\|\boldsymbol{y}\|_1} = \frac{c\sqrt{2(1-\rho)}}{\lambda_2}, \tag{A.32}$$

Thus the Theorem 4.2 is proved. □

A.4. Proof of Theorem 4.3

Proof Let $\hat{\mathcal{J}} = -\mathcal{J}$, according to the definition of (A.13), to prove that sparsity is established, just prove the following facts:

(1) For $\forall\ \boldsymbol{\theta} \in \omega$, let $\epsilon_n = \boldsymbol{\theta} - \bar{\boldsymbol{\theta}}$, it holds $\epsilon_n = \tilde{c}n^{-1/2}$, where $\tilde{c}$ is an arbitrary sufficiently large constant;

(2) For $\forall\ j = l+1, l+2, \cdots, p$, it holds

$$\frac{\partial \hat{\mathcal{J}}}{\partial \theta_j} =: \begin{cases} > 0, \ \text{if} \ -\epsilon_n < \theta_j < 0, \\ < 0, \ \text{if} \ 0 < \theta_j < \epsilon_n. \end{cases} \tag{A.33}$$

Therefore, it derives

$$\max_{\|\boldsymbol{\theta}\|_2 < \tilde{c}n^{-1/2}} \hat{\mathcal{J}}((\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)^{\mathrm{T}}; \mathcal{D}, \lambda) = \hat{\mathcal{J}}((\boldsymbol{\theta}_1, \boldsymbol{0})^{\mathrm{T}}; \mathcal{D}, \lambda). \tag{A.34}$$

The proof of the fact as follows. For

$$\frac{\partial \hat{\mathcal{J}}(\boldsymbol{\theta}; \mathcal{D}, \boldsymbol{\lambda})}{\partial \theta_j} = \frac{\partial \mathcal{L}(\boldsymbol{\theta}; \mathcal{D})}{\partial \theta_j} - \lambda_1 sign(\theta_j) - \lambda_2 \boldsymbol{\theta}^{\mathrm{T}}\mathcal{L}_j, \tag{A.35}$$

We have the expansion of Taylor series of the first term on the right side of (A.35), i.e.,

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta};\mathcal{D})}{\partial \theta_j} \approx \frac{\partial \mathcal{L}(\bar{\boldsymbol{\theta}};\mathcal{D})}{\partial \theta_j} + \sum_{l=1}^{p} \frac{\partial^2 \mathcal{L}(\bar{\boldsymbol{\theta}};\mathcal{D})}{\partial \theta_j \partial \theta_l}(\theta_l - \bar{\theta}_l) + \sum_{l=1}^{p}\sum_{k=1}^{p} \frac{\partial^3 \mathcal{L}(\bar{\boldsymbol{\theta}};\mathcal{D})}{\partial \theta_j \partial \theta_l \partial \theta_k}(\theta_l - \bar{\theta}_l)(\theta_k - \bar{\theta}_k), \quad \text{(A.36)}$$

with

$$\frac{1}{n}\frac{\partial \mathcal{L}(\bar{\boldsymbol{\theta}};\mathcal{D})}{\partial \theta_j} = \mathcal{O}(n^{-\frac{1}{2}}),$$
$$\frac{1}{n}\frac{\partial^2 \mathcal{L}(\bar{\boldsymbol{\theta}};\mathcal{D})}{\partial \theta_j \theta_l} = E\left[\frac{\partial^2 \mathcal{L}(\bar{\boldsymbol{\theta}};\mathcal{D})}{\partial \theta_j \partial \theta_l}\right] + \mathcal{O}(1), \quad \epsilon_n = \mathcal{O}(n^{-\frac{1}{2}}). \quad \text{(A.37)}$$

Combining with (A.37), (A.36) can be written as

$$\frac{\partial \hat{\mathcal{J}}(\boldsymbol{\theta};\mathcal{D},\boldsymbol{\lambda})}{\partial \theta_j} = n\left\{-\lambda_1 sign(\theta_j) - \lambda_2 \boldsymbol{\theta}^{\mathrm{T}}\mathcal{L}_j + \mathcal{O}(n^{-\frac{1}{2}})\right\}. \quad \text{(A.38)}$$

However,

$$\lim_{\boldsymbol{\theta}\to\mathbf{0}} \boldsymbol{\theta}^{\mathrm{T}}\mathcal{L}_j = 0, \quad \lim_{n\to\infty} n^{-\frac{1}{2}} = 0. \quad \text{(A.39)}$$

So the sign of the derivative $\frac{\partial \hat{\mathcal{J}}(\boldsymbol{\theta};\mathcal{D},\boldsymbol{\lambda})}{\partial \theta_j}$ depends entirely on the sign of $\boldsymbol{\theta}$. Thus the sparsity is proved.

Next we prove the asymptotic normality. By (4.1), (A.13) and (2.9), we have

$$\hat{\boldsymbol{\theta}} = \arg\min\left\{\sum_{i=1}^{n}\left[-y_i \boldsymbol{X}_i^T\boldsymbol{\theta} + \log(1+\exp(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\theta}))\right] + \lambda_1\sum_{j=1}^{p}|\theta_j| + \frac{\lambda_2}{2}\sum_{\substack{k\sim j\\ k\in V}}\left(\frac{\theta_k}{\sqrt{d_k}} - \frac{\theta_j}{\sqrt{d_j}}\right)^2 w_{kj}\right\}. \quad \text{(A.40)}$$

Let $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}} + \frac{\boldsymbol{\mu}}{\sqrt{n}}$, we define

$$\Psi(\boldsymbol{\mu}) = \sum_{i=1}^{n}\left[-y_i \boldsymbol{X}_i^T(\bar{\boldsymbol{\theta}} + \frac{\boldsymbol{\mu}}{\sqrt{n}}) + \log\left(1+\exp\left(\boldsymbol{X}_i^{\mathrm{T}}\left(\bar{\boldsymbol{\theta}} + \frac{\boldsymbol{\mu}}{\sqrt{n}}\right)\right)\right)\right] + \lambda_1\sum_{j=1}^{p}\left|\bar{\theta}_j + \frac{\mu_j}{\sqrt{n}}\right| + \frac{\lambda_2}{2}\sum_{\substack{k\sim j\\ k\in V}}\left(\frac{\bar{\theta}_k + \frac{\mu_k}{\sqrt{n}}}{\sqrt{d_k}} - \frac{\bar{\theta}_j + \frac{\mu_j}{\sqrt{n}}}{\sqrt{d_j}}\right)^2 w_{kj}, \quad \text{(A.41)}$$

then

$$\Psi(\mathbf{0}) = \sum_{i=1}^{n}\left[-y_i \boldsymbol{X}_i^T\bar{\boldsymbol{\theta}} + \log(1+\exp(\boldsymbol{X}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}}))\right] + \lambda_1\sum_{j=1}^{p}|\bar{\theta}_j| + \frac{\lambda_2}{2}\sum_{\substack{k\sim j\\ k\in V}}\left(\frac{\bar{\theta}_k}{\sqrt{d_k}} - \frac{\bar{\theta}_j}{\sqrt{d_j}}\right)^2 w_{kj}. \quad \text{(A.42)}$$

Let $g(\boldsymbol{\mu}) = \Psi(\boldsymbol{\mu}) - \Psi(\mathbf{0})$ be an auxiliary function, then

$$\begin{aligned} g(\boldsymbol{\mu}) &= \sum_{i=1}^{n}\left(-y_i \boldsymbol{X}_i^T(\bar{\boldsymbol{\theta}} + \frac{\boldsymbol{\mu}}{\sqrt{n}}) + y_i \boldsymbol{X}_i^T\bar{\boldsymbol{\theta}}\right) \\ &+ \sum_{i=1}^{n}\left[\log\left(1+\exp\left(\boldsymbol{X}_i^{\mathrm{T}}\left(\bar{\boldsymbol{\theta}} + \frac{\boldsymbol{\mu}}{\sqrt{n}}\right)\right)\right) - \log(1+\exp(\boldsymbol{X}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}}))\right] \\ &+ \lambda_1\sum_{j=1}^{p}\left(\left|\bar{\theta}_j + \frac{\mu_j}{\sqrt{n}}\right| - |\bar{\theta}_j|\right) \\ &+ \frac{\lambda_2}{2}\sum_{\substack{k\sim j\\ k\in V}}\left[\left(\frac{\bar{\theta}_k + \frac{\mu_k}{\sqrt{n}}}{\sqrt{d_k}} - \frac{\bar{\theta}_j + \frac{\mu_j}{\sqrt{n}}}{\sqrt{d_j}}\right)^2 w_{kj} - \left(\frac{\bar{\theta}_k}{\sqrt{d_k}} - \frac{\bar{\theta}_j}{\sqrt{d_j}}\right)^2 w_{kj}\right]. \end{aligned} \quad \text{(A.43)}$$

For the second term of the right side of (A.43), Using Taylor expansion and omitting the high-order terms, we have

$$\begin{aligned} &\sum_{i=1}^{n}\left[\log\left(1+\exp\left(\boldsymbol{X}_i^{\mathrm{T}}\left(\bar{\boldsymbol{\theta}} + \frac{\boldsymbol{\mu}}{\sqrt{n}}\right)\right)\right) - \log(1+\exp(\boldsymbol{X}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}}))\right] \\ &= \sum_{i=1}^{n}\log\left(\frac{1+\exp(\boldsymbol{X}_i^{\mathrm{T}}(\bar{\boldsymbol{\theta}} + \frac{\boldsymbol{\mu}}{\sqrt{n}}))}{1+\exp(\boldsymbol{X}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}})}\right) \\ &\approx \frac{1+\exp(\boldsymbol{X}_i^{\mathrm{T}}(\bar{\boldsymbol{\theta}} + \frac{\boldsymbol{\mu}}{\sqrt{n}}))}{1+\exp(\boldsymbol{X}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}})} - 1 \\ &= \frac{\exp(\boldsymbol{X}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}})(\exp(\boldsymbol{X}_i^{\mathrm{T}}\frac{\boldsymbol{\mu}}{\sqrt{n}}) - 1)}{1+\exp(\boldsymbol{X}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}})} \\ &\approx \frac{\exp(\boldsymbol{X}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}})}{1+\exp(\boldsymbol{X}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}})}(\boldsymbol{X}_i^{\mathrm{T}}\frac{\boldsymbol{\mu}}{\sqrt{n}}) \\ &= \frac{1}{\sqrt{n}}\frac{\exp(\boldsymbol{X}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}})}{1+\exp(\boldsymbol{X}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}})}(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\mu}). \end{aligned} \quad \text{(A.44)}$$

So (A.43) can be rewritten as

$$g(\boldsymbol{\mu}) = g_1(\boldsymbol{\mu}) + g_2(\boldsymbol{\mu}) + g_3(\boldsymbol{\mu}) + g_4(\boldsymbol{\mu}) + g_5(\boldsymbol{\mu}), \quad \text{(A.45)}$$

where

$$g_1(\boldsymbol{\mu}) = -\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\left(y_i - \frac{\exp(\boldsymbol{X}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}})}{1+\exp(\boldsymbol{X}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}})}\right)\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\mu}, \quad \text{(A.46)}$$

$$g_2(\boldsymbol{\mu}) = \frac{1}{2n}\sum_{i=1}^{n}\left(\frac{\exp^2(\boldsymbol{X}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}})}{(1+\exp(\boldsymbol{X}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}}))^2}\right)\boldsymbol{\mu}^{\mathrm{T}}(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{X}_i)\boldsymbol{\mu}, \quad \text{(A.47)}$$

$$g_3(\boldsymbol{\mu}) = \frac{1}{3n^{3/2}}\sum_{i=1}^{n}\left(\frac{\exp^3(\boldsymbol{X}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}})}{(1+\exp(\boldsymbol{X}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}}))^3}\right)(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\mu})^3, \quad \text{(A.48)}$$

$$g_4(\boldsymbol{\mu}) = \lambda_1^{(n)}\sum_{j=1}^{p}\left(\left|\bar{\theta}_j + \frac{\mu_j}{\sqrt{n}}\right| - |\bar{\theta}_j|\right), \quad \text{(A.49)}$$

$$g_5(\boldsymbol{\mu}) = \frac{\lambda_2^{(n)}}{2} \sum_{\substack{k\sim j \\ k\in V}} \left\{ \left[\left(\frac{\bar{\theta}_k}{\sqrt{d_k}} - \frac{\bar{\theta}_j}{\sqrt{d_j}} \right) + \left(\frac{\mu_k}{\sqrt{nd_k}} - \frac{\mu_j}{\sqrt{nd_j}} \right) \right]^2 w_{kj} - \left(\frac{\bar{\theta}_k}{\sqrt{d_k}} - \frac{\bar{\theta}_j}{\sqrt{d_j}} \right)^2 w_{kj} \right\}. \tag{A.50}$$

Secondly, we analyze the convergence of them item by item. For $g_1(\boldsymbol{\mu})$, from the properties of the exponential family distribution [86], we have

$$E\left\{ \left(y_i - \frac{\exp(\boldsymbol{X}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}})}{1+\exp(\boldsymbol{X}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}})} \right) \right\} = 0, \tag{A.51}$$

and

$$Var\left\{ \left(y_i - \frac{\exp(\boldsymbol{X}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}})}{1+\exp(\boldsymbol{X}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}})} \right) \right\} = \boldsymbol{\mu}^{\mathrm{T}}\mathcal{C}\boldsymbol{\mu}, \tag{A.52}$$

Thus by the Central Limit Theorem (CLT) [87, 88], we obtain

$$g_1(\boldsymbol{\mu}) \xrightarrow{d} -\boldsymbol{\mu}^{\mathrm{T}}\mathcal{W}, \tag{A.53}$$

where $\mathcal{W} \sim N(\mathbf{0}, \mathcal{C})$.

For $g_2(\boldsymbol{\mu})$, by the Law of Large Numbers (LLN) [89, 90], we have

$$g_2(\boldsymbol{\mu}) \xrightarrow{p} \frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\mathcal{C}\boldsymbol{\mu}. \tag{A.54}$$

For $g_3(\boldsymbol{\mu})$, by the condition, we know that

$$3\sqrt{n}\, g_3(\boldsymbol{\mu}) = \frac{1}{n}\sum_{i=1}^{n} \left(\frac{\exp^2(\boldsymbol{X}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}})}{(1+\exp(\boldsymbol{X}_i^{\mathrm{T}}\bar{\boldsymbol{\theta}}))^2} \right) \boldsymbol{\mu}^{\mathrm{T}}(\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{X}_i)\boldsymbol{\mu} \le \sum_{i=1}^{n} M(\boldsymbol{X}_i^{\mathrm{T}})|\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\mu}|^3 \xrightarrow{p} E\{M(\boldsymbol{X})|\boldsymbol{X}_i^{\mathrm{T}}\boldsymbol{\mu}|^3\} < \infty, \tag{A.55}$$

So

$$g_3(\boldsymbol{\mu}) \xrightarrow{p} 0. \tag{A.56}$$

For $g_4(\boldsymbol{\mu})$, with finite dimensional convergence holding trivially, we also have

$$g_4(\boldsymbol{\mu}) \to \lambda_1^* \sum_{j=1}^{p} \{\mu_j sign(\bar{\theta}_j) I(\bar{\theta}_j \neq 0) + |\mu_j| I(\bar{\theta}_j = 0)\}, \tag{A.57}$$

So

$$g_4(\boldsymbol{\mu}) \xrightarrow{p} 0. \tag{A.58}$$

For $g_5(\boldsymbol{\mu})$, we have

$$g_5(\boldsymbol{\mu}) \to \lambda_2^* \sum_{\substack{k\sim j \\ k\in V}} \left(\frac{\bar{\theta}_k}{\sqrt{d_k}} - \frac{\bar{\theta}_j}{\sqrt{d_j}} \right) \left(\frac{\mu_k}{\sqrt{d_k}} - \frac{\mu_j}{\sqrt{d_j}} \right) w_{kj}. \tag{A.59}$$

So

$$g_5(\boldsymbol{\mu}) \xrightarrow{p} 0. \tag{A.60}$$

Thus, considering (A.53), (A.54), (A.56), (A.58) and (A.60), according to Slutsky theorem [91, 92], we obtain

$$g(\boldsymbol{\mu}) \xrightarrow{p} \frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\mathcal{C}\boldsymbol{\mu} - \boldsymbol{\mu}^{\mathrm{T}}\mathcal{W}. \tag{A.61}$$

where $g(\boldsymbol{\mu})$ is a convex function with respect to $\boldsymbol{\mu}$ and has a unique minimum point. Let $g'(\boldsymbol{\mu}) = 0$, it derives $\boldsymbol{\mu} = \mathcal{C}^{-1}\mathcal{W}$. So the minimum point of $g(\boldsymbol{\mu})$ is $(\mathcal{C}^{-1}\mathcal{W}, \mathbf{0})^{\mathrm{T}}$. Thus the asymptotic normality is proved. □

A.5. Proof of Theorem 4.4

Proof Let $(\boldsymbol{\theta}^*, \boldsymbol{u}^*)$ be an optimal solution of the equivalent problem (3.1), given a sufficiently small positive number ε, by the assumption mentioned above, there is an $(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{u}}) \in \mathrm{dom}\phi$ such that

$$\Phi(\boldsymbol{\theta}^*, \boldsymbol{u}^*) + \varepsilon > \Phi(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{u}}).$$

Then for any $\mu > 0$, it holds that

$$\begin{aligned} &\Phi(\boldsymbol{\theta}^*, \boldsymbol{u}^*) + \varepsilon + \phi_\mu(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{u}}) \\ &> \Phi(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{u}}) + \phi_\mu(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{u}}) \\ &\ge \Phi(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu) + \phi_\mu(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu) \\ &= Q_\mu(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu). \end{aligned} \tag{A.62}$$

Taking the limit as $\mu \to 0^+$ for (A.62), and by (3.5), it follows that

$$\Phi(\boldsymbol{\theta}^*, \boldsymbol{u}^*) + \varepsilon > \lim_{\mu\to 0^+} Q_\mu(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu). \tag{A.63}$$

Since (A.63) holds for any $\varepsilon > 0$, it drives

$$\Phi(\boldsymbol{\theta}^*, \boldsymbol{u}^*) \ge \lim_{\mu\to 0^+} Q_\mu(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu). \tag{A.64}$$

However, by the definition of barrier function ϕ_μ, it is clear that $\phi_\mu(\boldsymbol{\theta}, \boldsymbol{u}) \ge 0$, $(\boldsymbol{\theta}, \boldsymbol{u}) \in \mathrm{dom}\,\phi_\mu$. Then for $\mu \ge 0$, we have

$$\begin{aligned} Q_\mu(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu) &= \inf\{\Phi(\boldsymbol{\theta}, \boldsymbol{u}) + \phi_\mu(\boldsymbol{\theta}, \boldsymbol{u}) | (\boldsymbol{\theta}, \boldsymbol{u}) \in \mathrm{dom}\phi_\mu\} \\ &\ge \inf\{\Phi(\boldsymbol{\theta}, \boldsymbol{u}) | (\boldsymbol{\theta}, \boldsymbol{u}) \in \mathrm{dom}\phi_\mu\} \\ &\ge \inf\{\Phi(\boldsymbol{\theta}, \boldsymbol{u}) | (\boldsymbol{\theta}, \boldsymbol{u}) \in \mathrm{dom}\Phi\}. \end{aligned} \tag{A.65}$$

So it holds

$$\Phi(\boldsymbol{\theta}^*, \boldsymbol{u}^*) \leq \lim_{\mu \to 0^+} Q_\mu(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu). \tag{A.66}$$

By (A.64) and (A.66), we have

$$\Phi(\boldsymbol{\theta}^*, \boldsymbol{u}^*) = \lim_{\mu \to 0^+} Q_\mu(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu). \tag{A.67}$$

Combining with the definition of $Q_\mu(\boldsymbol{\theta}_\mu, \boldsymbol{u})$ and considering that $(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu)$ is feasible to the equivalent problem (3.1) for $\mu \geq 0$, it follows that

$$\begin{aligned} Q_\mu(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu) &= \Phi(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu) + \phi_\mu(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu) \\ &\geq \Phi(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu) \\ &\geq \Phi(\boldsymbol{\theta}^*, \boldsymbol{u}^*). \end{aligned} \tag{A.68}$$

Taking the limit as $\mu \to 0^+$ for (A.68), it follows that

$$\lim_{\mu \to 0^+} Q_\mu(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu) = \lim_{\mu \to 0^+} \left\{ \Phi(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu) + \phi_\mu(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu) \right\}. \tag{A.69}$$

Combining with (A.67), it derives

$$\lim_{\mu \to 0^+} \left\{ \Phi(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu) + \phi_\mu(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu) \right\} = \Phi(\boldsymbol{\theta}^*, \boldsymbol{u}^*). \tag{A.70}$$

Considering the continuity of function $\Phi(\cdot)$ in (A.70), it holds that

$$\lim_{\mu \to 0^+} \Phi(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu) = \Phi(\boldsymbol{\theta}^*, \boldsymbol{u}^*). \tag{A.71}$$

Combining (A.70) and (A.71), it gives that

$$\lim_{\mu \to 0^+} \phi_\mu(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu) = 0. \tag{A.72}$$

Furthermore, if $\{(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu)\}$ has a convergent subsequence with limit $(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{u}})$, then $\Phi(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{u}}) = \Phi(\boldsymbol{\theta}^*, \boldsymbol{u}^*)$. Since $(\boldsymbol{\theta}_\mu, \boldsymbol{u}_\mu)$ is the feasible solution to the equivalent problem (3.1) for each $\mu \geq 0$, it follows that $(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{u}})$ is also feasible and optimal. □

A.6. Proof of Theorem 4.5

Proof Suppose that $(\boldsymbol{\theta}^*, \boldsymbol{u}^*, \boldsymbol{z}^{\theta^*}, \boldsymbol{z}^{u^*})$ is a limit point of $\{(\boldsymbol{\theta}_k, \boldsymbol{u}_k, \boldsymbol{z}_k^\theta, \boldsymbol{z}_k^u)\}$ and the subsequence $\{(\boldsymbol{\theta}_k, \boldsymbol{u}_k, \boldsymbol{z}_k^\theta, \boldsymbol{z}_k^u)\}$ converges to $(\boldsymbol{\theta}^*, \boldsymbol{u}^*, \boldsymbol{z}^{\theta^*}, \boldsymbol{z}^{u^*})$. When $(\boldsymbol{\theta}_k, \boldsymbol{u}_k, \boldsymbol{z}_k^\theta, \boldsymbol{z}_k^u)$ is the solution of (3.12), taking the limit as $k \to$ inf to both sides of each equation of (3.12), it derives

$$\begin{cases} X^{\mathrm{T}}(\boldsymbol{f} - \boldsymbol{y}) + \lambda_2 \mathcal{L}\boldsymbol{\theta}^* + \boldsymbol{z}^{\theta^*} = 0, \\ \lambda_1 \mathbf{1} - \boldsymbol{z}^{u^*} = 0, \\ Z^\theta \Sigma \mathbf{1} - 2\mu\boldsymbol{\theta}^* = 0, \\ Z^u \Sigma \mathbf{1} - 2\mu\boldsymbol{u}^* = 0. \end{cases} \tag{A.73}$$

This is because of the fact

$$\lim_{k \to \infty} \{\triangle\boldsymbol{\theta}_k,\ \triangle\boldsymbol{u}_k,\ \triangle\boldsymbol{z}_k^\theta,\ \triangle\boldsymbol{z}_k^u)\} = \{0,\ 0,\ 0,\ 0\}. \tag{A.74}$$

Thus $\{(\boldsymbol{\theta}^*, \boldsymbol{u}^*, \boldsymbol{z}^{\theta^*}, \boldsymbol{z}^{u^*})\}$ satisfies the first-order optimal condition (3.10). □

Appendix B

Figure 8 shows the general framework for biomarkers discovery from high-throughput transcriptomics data by CNet-RLR model. The biomarker discovery experiment is implemented in the dataset of HiSeqV2. As shown in Fig. 8, we firstly download the dataset from TCGA and obtain 16,245 genes by integrating with the genes have prior regulatory interactions deposited in RegNetwork [36].

Secondly, we select 360 differentially expressed genes (DEGs) with adj.P.Val$<$ 0.05 and |logFC| $>$ 2 by the empirical Bayesian method [93]. The prior network of structured features are extracted from RegNetwork [36]. The full list of DEGs are shown in Table S3 and the gene network is shown in Figure S2 in the supplementary file. They will be implemented for identifying biomarkers using the proposed CNet-RLR model. To ensure the balance of positive and negative samples in the classification, we randomly take 24 samples from 117 disease samples without replacement and match them with 24 normal samples. This induces a new integrated dataset with the same number of normal and UCEC samples.

Thirdly, for training and testing purpose, the new generated dataset is divided into two subsets. We randomly select 75% samples in the entire dataset for training, and the remaining 25% samples as the internal validation dataset for identified biomarkers. The optimal parameter λ_1 and λ_2 are determined through 5-fold CV to obtain the minimum value of the objective function in CNet-RLR model. We select the features/variables/genes whose coefficients $\theta_i > 1e-6$ ($\theta_i \leq 1e-6$ is regarded as 0) from the solutions of CNet-RLR model. Then we evaluate their classification performances on the internal validation dataset. Thus a subset of feature genes, 27 in total, is obtained. We find that these identified biomarkers are connected in the form of a network.

Finally, we perform functional enrichment analysis on the identified UCEC biomarkers and make the external validation to them on the independent dataset GSE63678. We get the AUC value of classification in the external validation using the trained LR classifier on the discovery dataset HiSeqV2. We also compared the results with the other RLR models and graph embeded deep learning model, i.e., Lasso-RLR, Enet-RLR, Net-RLR, ABNet-RLR and GEDFN model. The features selected by CNet-RLR achieve better classification performance (e.g., AUC = 0.900) on the external validation dataset. The results prove the effectiveness and advantage of CNet-RLR in biomarker discovery by feature selection.

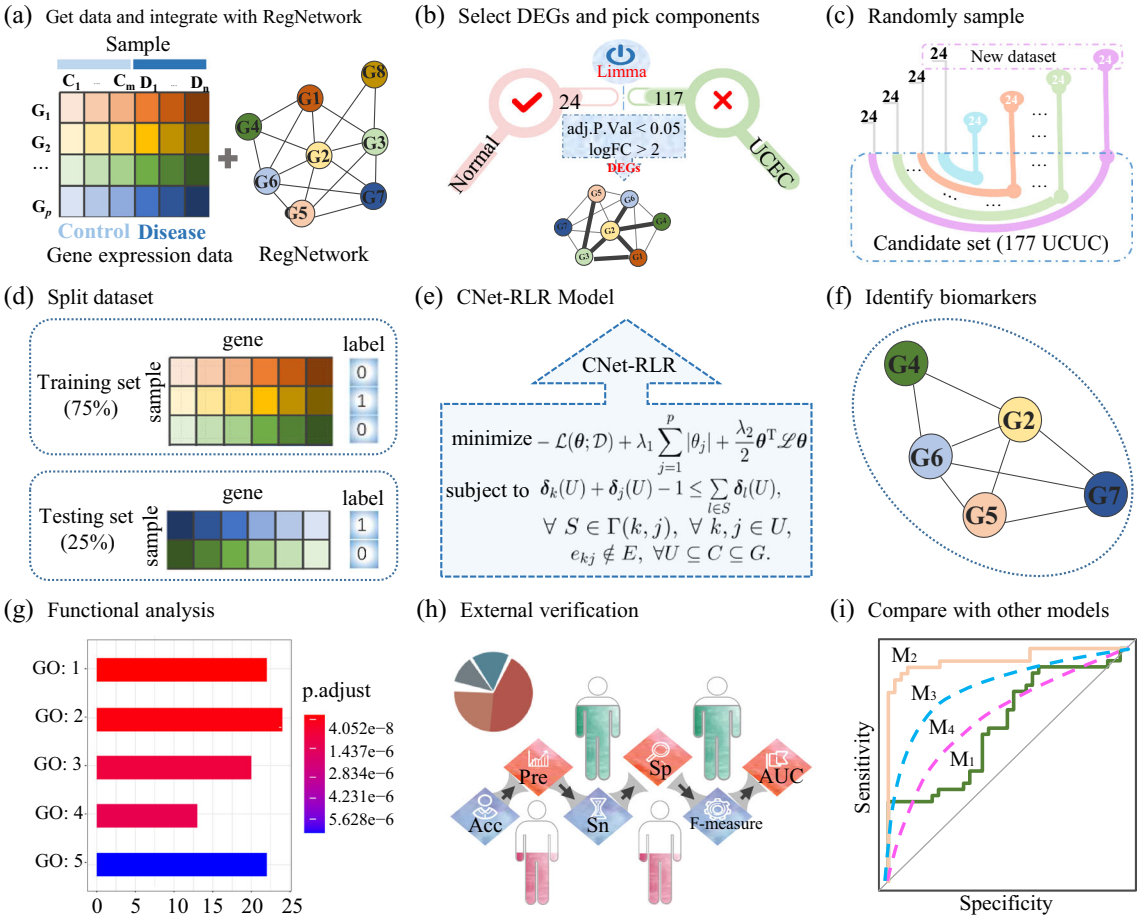

Fig. 8 The framework of identifying predictive cancer biomarkers by CNet-RLR

References

1. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3(3):1157–1182
2. Bommert A, Sun X, Bischl B, Rahnenführer J, Lang M (2020) Benchmark for filter methods for feature selection in high-dimensional classification data. Comput Stat Data Anal 143:106839
3. Cui X, Li Y, Fan J, Wang T (2021) A novel filter feature selection algorithm based on relief. Appl Intell:1–19
4. Mohanty BP, Mohanty D, Mitra T, Ganguly S, Mahanty A, Mohanty S, Karunakaran D Big data science and omics technology. In: Fisheries biology: New approaches and changing perspectives, first edition, Chapter 25. Narendra Publishing House, Delhi, pp 251–270
5. Moncada R, Barkley D, Wagner F, Chiodin M, Devlin JC, Baron M, Hajdu CH, Simeone DM, Yanai I (2020) Integrating microarray-based spatial transcriptomics and single-cell rna-seq reveals tissue architecture in pancreatic ductal adenocarcinomas. Nat Biotechnol 38(3):333–342
6. Yang Q, Li B, Tang J, Cui X, Wang Y, Li X, Hu J, Chen Y, Xue W, Lou Y et al (2020) Consistent gene signature of schizophrenia identified by a novel feature selection strategy from comprehensive sets of transcriptomic data. Brief Bioinform 21(3):1058–1068
7. Wu Y, Wu Q, Dey N, Sherratt S (2020) Learning models for semantic classification of insufficient plantar pressure images. Int J Interact Multimed Artif Intell 6(1):51–61
8. Li X, Li R, Xia Z, Xu C (2020) Distributed feature screening via componentwise debiasing. J Mach Learn Res 21(24):1–32
9. Liu Z-P (2016) Identifying network-based biomarkers of complex diseases from high-throughput data. Biomark Med 10(6):633–650
10. Cheng W, Zhang X, Guo Z, Yu S, Wang W (2014) Graph-regularized dual lasso for robust eqtl mapping. Bioinformatics 30(12):i139–i148
11. Brito-Pacheco C, Brito-Loeza C, Martin-Gonzalez A (2020) A regularized logistic regression based model for supervised learning. J Algorithm Comput Technol 14:1–9
12. Kumar P, Dayal M, Khari M, Fenza G, Gallo M (2021) Nsl-bp: A meta classifier model based prediction of amazon product reviews. Int J Interact Multimed Artif Intell 6(6):95–103
13. Karlos S, Kostopoulos G, Kotsiantis S (2020) A soft-voting ensemble based co-training scheme using static selection for binary classification problems. Algorithms 13(1):1–19
14. Hans R, Kaur H (2020) Binary multi-verse optimization (bmvo) approaches for feature selection. Int J Interact Multimed Artif Intell 6(1):91–106
15. Hastie T, Tibshirani R, Wainwright M (2015) Statistical Learning with Sparsity: The Lasso and Generalizations. CRC press
16. Ya Arsenin V, Krianev AV (2020) Generalized maximum likelihood method and its application for solving ill-posed

problems. In: Ill-posed problems in natural sciences. de gruyter, pp 1–12
17. Tikhonov AN, Arsenin VY (1977) Solutions of ill-posed problems, New York, pp 1–30
18. Li L, Liu Z-P (2020) Biomarker discovery for predicting spontaneous preterm birth from gene expression data by regularized logistic regression. Comput Struct Biotechnol J 18:3434–3446
19. Hoerl AE, Kennard RW (1970) Ridge regression: Biased estimation for nonorthogonal problems. Technometrics 12(1):55–67
20. Tibshirani R (1996) Regression shrinkage and selection via the lasso. J R Stat Soc Ser B (Stat Methodol) 58(1):267–288
21. Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. J R Stat Soc Ser B (Stat Methodol) 67(2):301–320
22. Yang L, Qian Y (2016) A sparse logistic regression framework by difference of convex functions programming. Appl Intell 45(2):241–254
23. Liu Z, Sun F, McGovern DP (2017) Sparse generalized linear model with l_0 approximation for feature selection and prediction with big omics data. BioData Mining 10(1):1–12
24. Xu Z, Zhang H, Wang Y, Chang X, Liang Y (2010) $l_{1/2}$ regularization. Sci China Inf Sci 53(6):1159–1169
25. Fan J, Li R (2001) Variable selection via nonconcave penalized likelihood and its oracle properties. J Amer Stat Assoc 96(456):1348–1360
26. Zhang C-H et al (2010) Nearly unbiased variable selection under minimax concave penalty. Ann Stat 38(2):894–942
27. Liang X, Jacobucci R (2020) Regularized structural equation modeling to detect measurement bias: Evaluation of lasso, adaptive lasso, and elastic net. Struct Equ Model Multidiscip J 27(5):722–734
28. Breheny P, Huang J (2011) Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. Ann Appl Stat 5(1):232–253
29. Knight K, Fu W (2000) Asymptotics for lasso-type estimators. Ann Stat 28(5):1356–1378
30. Li C, Li H (2008) Network-constrained regularization and variable selection for analysis of genomic data. Bioinformatics 24(9):1175–1182
31. Zhang W, Wan Y-w, Allen GI, Pang K, Anderson ML, Liu Z (2013) Molecular pathway identification using biological network-regularized logistic models. BMC Genomics 14(S8):1–8
32. Sun H, Lin W, Feng R, Li H (2014) Network-regularized high-dimensional cox regression for analysis of genomic data. Stat Sin 24(3):1433–1459
33. Ng B, Siless V, Varoquaux G, Poline J-B, Thirion B, Abugharbieh R (2012) Connectivity-informed sparse classifiers for fmri brain decoding. In: 2012 Second international workshop on pattern recognition in neuroimaging. IEEE, pp 101–104
34. Liu C, Wong HS (2017) Structured penalized logistic regression for gene selection in gene expression data analysis. IEEE/ACM Trans Comput Biol Bioinform 16(1):312–321
35. Li C, Xuan J, Riggins RB, Clarke R, Wang Y (2011) Identifying cancer biomarkers by network-constrained support vector machines. BMC Syst Biol 5(1):1–20
36. Liu Z-P, Wu C, Miao H, Wu H (2015) Regnetwork: an integrated database of transcriptional and post-transcriptional regulatory networks in human and mouse. Database 2015:1–12
37. Chung FRK, Graham FC (1997) Spectral graph Theory. Number 92. American Mathematical Society
38. Newman M (2018) Networks. Oxford University Press, Oxford
39. Li C, Li H (2010) Variable selection and regression analysis for graph-structured covariates with an application to genomics. Ann Appl Stat 4(3):1498–1516
40. Bapat RB (2010) Graphs and Matrices, vol 27. Springer
41. Franklin JN (2012) Matrix theory. Courier Corporation
42. Binder H, Schumacher M (2008) Comment on 'network-constrained regularization and variable selection for analysis of genomic data'. Bioinformatics 24(21):2566–2568
43. Li C, Li H (2008) In response to comment on 'network-constrained regularization and variable selection for analysis of genomic data'. Bioinformatics 24(21):2569–2569
44. Mei Q, Cai D, Zhang D, Zhai C (2008) Topic modeling with network regularization. In: Proceedings of the 17th International Conference on World Wide Web. WWW 2008, Beijing, pp 101–110
45. Zhou J, Chen J, Ye J (2011) Malsar: Multi-task learning via structural regularization. Arizona State University, pp 21
46. Wu M-Y, Zhang X-F, Dai D-Q, Le O-Y, Zhu Y, Yan H (2016) Regularized logistic regression with network-based pairwise interaction for biomarker identification in breast cancer. BMC Bioinform 17(1):1–18
47. Min W, Liu J, Zhang S (2016) Network-regularized sparse logistic regression models for clinical risk prediction and biomarker discovery. IEEE/ACM Trans Comput Biol Bioinform 15(3):944–953
48. Carvajal R, Constantino M, Goycoolea M, Vielma JP, Weintraub A (2013) Imposing connectivity constraints in forest planning models. Oper Res 61(4):824–836
49. Kong Y, Yu T (2018) A graph-embedded deep feedforward network for disease outcome classification and feature selection using gene expression data. Bioinformatics 34(21):3727–3737
50. Saleem N, Khattak MI (2020) Deep neural networks for speech enhancement in complex-noisy environments. Int J Interact Multimed Artif Intell 6(1):84–90
51. Álvarez-Miranda E, Markus SA (2017) Relax-and-cut framework for large-scale maximum weight connected subgraph problems. Comput Oper Res 87:63–82
52. Althaus E, Blumenstock M, Disterhoft A, Hildebrandt A, Krupp M (2014) Algorithms for the maximum weight connected k-induced subgraph problem. In: International conference on combinatorial optimization and applications. Springer, pp 268–282
53. Li Q, Chen W, Liu S, Tong L (2016) Structural topology optimization considering connectivity constraint. Struct Multidiscip Optim 54(4):971–984
54. Cawley GC, Talbot NLC (2010) On over-fitting in model selection and subsequent selection bias in performance evaluation. J Mach Learn Res 11(1):2079–2107
55. Liang S, Khoo Y, Yang H (2021) Drop-activation: implicit parameter reduction and harmonious regularization. Commun Appl Math Comput 3(2):293–311
56. Qiao X (2014) Variable selection using l_q penalties. Wiley Interdiscip Rev Comput Stat 6(3):177–184
57. Koh K, Kim S-J, Boyd S (2007) An interior-point method for large-scale l_1-regularized logistic regression. J Mach Learn Res 8(7):1519–1555
58. Boyd S, Cheriyan J, Haddadan A, Ibrahimpur S (2021) A 2-approximation algorithm for flexible graph connectivity. arXiv:2102.03304
59. Zhou D, Schölkopf B (2006) Discrete regularization. MIT press
60. Bougleux S, Elmoataz A, Melkemi M (2009) Local and nonlocal discrete regularization on weighted graphs for image and mesh processing. Int J Comput Vis 84(2):220–236
61. Zhou D, Bousquet O, Lal TN, Weston J, Schölkopf B (2004) Learning with local and global consistency. In: Advances in neural information processing systems, pp 321–328
62. Golub GH, Van loan CF (2013) Matrix Computations, 4th edn. Johns Hopkins Universtiy Press

63. Wang Y, Buchanan A, Butenko S (2017) On imposing connectivity constraints in integer programs. Math Program 166(1-2):241–271
64. Grötschel M, Monma CL (1990) Integer polyhedra arising from certain network design problems with connectivity constraints. SIAM J Discret Math 3(4):502–523
65. West DB et al (2001) Introduction to graph theory, vol 2. Prentice Hall, Upper Saddle River
66. Scott Provan J, Shier DR (1996) A paradigm for listing (s, t)-cuts in graphs. Algorithmica 15(4):351–372
67. Rao MM (2018) Measure theory and integration. CRC Press
68. Yao L, Zeng F, Li D-H, Chen Z-G (2017) Sparse support vector machine with l_p penalty for feature selection. J Comput Sci Technol 32(1):68–77
69. Fathi-Hafshejani S, Moaberfard Z (2020) An interior-point algorithm for linearly constrained convex optimization based on kernel function and application in non-negative matrix factorization. Optim Eng 21(3):1019–1051
70. Yao L, Zhang X, Li D-H, Zeng F, Chen H (2014) An interior point method for $l_{1/2}$-SVM and application to feature selection in classification. J Appl Math 2014:1–16
71. Wächter A, Biegler LT (2006) On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math Program 106(1):25–57
72. Mockus J (2012) Bayesian approach to global optimization: theory and applications, vol 37. Springer Science & Business Media
73. Tomczak K, Czerwińska P, Wiznerowicz M (2015) The cancer genome atlas (tcga): an immeasurable source of knowledge. Contemp Oncol 19(1A):A68–A77
74. Goldman M, Craft B, Swatloski T, Cline M, Morozova O, Diekhans M, Haussler D, Zhu J (2015) The ucsc cancer genomics browser: update 2015. Nucleic Acids Res 43(D1):D812–D817
75. Pappa KI, Polyzos A, Jacob-Hirsch J, Amariglio N, Vlachos GD, Loutradis D, Anagnou NP (2015) Profiling of discrete gynecological cancers reveals novel transcriptional modules and common features shared by other cancer types and embryonic stem cells. PLoS One 10(11):1–20
76. Carbon S, Douglass E, Good BM, Unni DR, Harris NL, Mungall CJ, Basu S, Chisholm RL, Dodson RJ, Hartline E et al (2021) The gene ontology resource: enriching a gold mine. Nucleic Acids Res 49(D1):D325–D334
77. Zhou Y, Zhou B, Pache L, Chang M, Khodabakhshi AH, Tanaseichuk O, Benner C, Chanda SK (2019) Metascape provides a biologist-oriented resource for the analysis of systems-level datasets. Nat Commun 10(1):1–10
78. Kandoth C, McLellan MD, Vandin F, Ye K, Niu B, Lu C, Xie M, Zhang Q, McMichael JF, Wyczalkowski MA et al (2013) Mutational landscape and significance across 12 major cancer types. Nature 502(7471):333–339
79. Li Q, Lei Y, Du W (2018) A novel target of p53, tcf21, can respond to hypoxia by mapk pathway inactivation in uterine corpus endometrial carcinoma. DNA Cell Biol 37(5):473–480
80. Zhang L, Wan Y, Yi J, Zhang Z, Shu S, Cheng W, Lang J (2019) Overexpression of bp1, an isoform of homeobox gene dlx4, promotes cell proliferation, migration and predicts poor prognosis in endometrial cancer. Gene 707:216–223
81. Wang X, Chen T (2020) Cul4a regulates endometrial cancer cell proliferation, invasion and migration by interacting with csn6. Mol Med Rep 23(1):1–9
82. Mello AC, Freitas M, Coutinho L, Falcon T, Matte U (2020) Machine learning supports long noncoding rnas as expression markers for endometrial carcinoma. BioMed Res Int 2020(10):1–12
83. Wang J, Huang Q, Liu Z-P, Wang Y, Wu L-Y, Chen L, Zhang X-S (2011) Noa: a novel network ontology analysis method. Nucleic Acids Res 39(13):e87–e98
84. Hanahan D, Weinberg RA (2000) The hallmarks of cancer. Cell 100(1):57–70
85. Zhu Y, Shen X, Pan W (2009) Network-based support vector machine for classification of microarray samples. BMC Bioinforma 10(1):1–11
86. Jamal F, Chesneau C, Elgarhy M (2020) Type ii general inverse exponential family of distributions. J Stat Manag Syst 23(3):617–641
87. de Jong P (1987) A central limit theorem for generalized quadratic forms. Probab Theory Relat Fields 75(2):261–277
88. Brosamler GA (1988) An almost everywhere central limit theorem. In: Mathematical Proceedings of the Cambridge Philosophical Society, vol 104. Cambridge University Press, pp 561–574
89. Hsu P-L, Robbins H (1947) Complete convergence and the law of large numbers. Proc Natl Acad Sci U S A 33(2):25
90. Judd KL (1985) The law of large numbers with a continuum of iid random variables. J Econ Theory 35(1):19–25
91. Ressel P (1982) A topological version of slutsky's theorem. Proc Am Math Soc 85(2):272–274
92. Delbaen F (1998) A remark on slutsky's theorem. In: Séminaire de probabilités XXXII. Springer, pp 313–315
93. Evan Johnson W, Li C, Rabinovic A (2007) Adjusting batch effects in microarray expression data using empirical bayes methods. Biostatistics 8(1):118–127

Lingyu Li received the B.S and M.S degrees in mathematics from Shandong Normal University in 2016 and 2019, respectively. She is a Ph.D. candidate at Shandong University. Her research interests include bioinformatics and machine learning.

Zhi-Ping Liu received the B.S and M.S degrees in mathematics from Shandong University in 2002 and 2005, respectively, and a Ph.D. from Academy of Mathematics and Systems Science, Chinese Academy of Sciences in 2008. He formerly worked as an associate professor at the Key Laboratory of Systems Biology, Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences. Currently, he is a professor of biomedical informatics at Shandong University, China. His research interests are computational biology, bioinformatics and machine learning.