Chen *et al. Journal of Statistical Distributions and Applications* (2019) 6:6
https://doi.org/10.1186/s40488-019-0094-2

Journal of Statistical Distributions and Applications

METHODOLOGY Open Access

Multiclass analysis and prediction with network structured covariates

Li-Pang Chen[1†], Grace Y. Yi[1*†], Qihuang Zhang[1†] and Wenqing He[2†]

*Correspondence:
yyi@uwaterloo.ca
†Li-Pang Chen and Grace Y. Yi contributed equally to this work.
†Qihuang Zhang and Wenqing He participate in the project with equal contributions.
[1]Department of Statistics and Actuarial Science, University of Waterloo, 200 University Ave W, N2L 3G1 Waterloo, Canada
Full list of author information is available at the end of the article

Abstract

Technological advances associated with data acquisition are leading to the production of complex structured data sets. The recent development on classification with multiclass responses makes it possible to incorporate the dependence structure of predictors. The available methods, however, are hindered by the restrictive requirements. Those methods basically assume a common network structure for predictors of all subjects without taking into account the heterogeneity existing in different classes. Furthermore, those methods mainly focus on the case where the distribution of predictors is normal. In this paper, we propose classification methods which address these limitations. Our methods are flexible in handling possibly class-dependent network structures of variables and allow the predictors to follow a distribution in the exponential family which includes normal distributions as a special case. Our methods are computationally easy to implement. Numerical studies are conducted to demonstrate the satisfactory performance of the proposed methods.

Keywords: F-score, Logistic regression model, Multiclassification, Network structure

Introduction

In contemporary statistical inference and machine learning theory, classification and prediction are of great importance and many approaches have been proposed. Those methods typically include the support vector machine (SVM), linear discriminant analysis (LDA), and K-nearest neighbors (KNN) (Hastie et al. 2008; James et al. 2017). These methods have widespread applications and their extensions to accommodating complex settings have been proposed. For example, Lee and Lee (2003) studied multicategory support vector machines for classification of multiple types of cancer. Cristianini and Shawe-Taylor (2000) presented comprehensive discussions of SVM methods. Guo et al. (2007) discussed the LDA method and its application in microarray data analysis. Safo and Ahn (2016) considered the multiclass analysis by performing the generalized sparse linear discriminant analysis. Regarding analysis of multiclass classification problems, Bagirov et al. (2003) proposed a new algorithm for multiclass cancer data. Bicciato et al. (2003) presented disjoint models for multiclass cancer analysis using the principal component technique. Liu et al. (2005) proposed the genetic algorithm (GA)-based algorithm to carry out multiclass cancer classification.

Recent development on classification further incorporates the dependence structure of predictors. For example, Cetiner and Akgul (2014) developed a graphical-model-based method for the multi-label classification. Zhu and Pan (2009) proposed the

network-based support vector machine for classification of microarray samples for binary classification. Zi et al. (2016) discussed identification of rheumatoid arthritis-related genes by using network-based support vector machine. Cai et al. (2018) considered the network linear discriminant analysis. Huttenhower et al. (2007) proposed the nearest neighbor network approach. In the Bayesian paradigm, various classification approaches with network-structures accommodated have been explored, such as Bielza et al. (2011), Miguel Hernández-Lobato et al. (2011), Baladanddayuthapani et al. (2014), and Peterson et al. (2015).

Although there have been methods handling network structures in classification, those methods basically assume a common network structure for predictors of all subjects without taking into account of possible heterogeneity for different classes. To overcome those shortcomings, in this paper we propose classification methods with possibly class-dependent network structures of predictors taken into account. Our methods utilize the graphical model theory and allow the predictors to follow an exponential family distribution, instead of a restrictive normal distribution. Furthermore, we develop a prediction criterion for multiclass classification which accommodates pairwise dependence structures among the predictors. Our methods facilitate informative predictors with pairwise dependence structures into classification procedures, and they are computationally easy to implement.

The remainder of the paper is organized as follows. In "Data structure and framework" section, we introduce the data structure and review a convenient multiclass classification method for simple settings. In "Classification with predictor graphical structures accommodated" section, we describe the basics of graphical model theory and propose two methods for multiclass classification to accommodate network structures of predictors. In "Evaluation of the performance" section, we describe the criteria for evaluating the performance of the proposed methods, and briefly review several competing classification methods for comparisons. In "Numerical studies" section, we conduct simulation studies to assess the performance of the proposed methods, and apply the proposed methods to analyze a real dataset for illustration. A general discussion is presented in the last section.

Data structure and framework

In this section, we present the data structure with multiclass responses and introduce the basic notation.

Notation

Suppose the data of n subjects come from I classes, where I is an integer no smaller than 2 and the classes are free of order, i.e., they are nominal. Let n_i be the class size in class i with $i = 1, \cdots, I$, and hence $n = \sum_{i=1}^{I} n_i$. Define $Y_{ik} = i$ for class $i = 1, \cdots, I$ and subject $k = 1, \cdots, n_i$, and let $Y = \left(Y_{11}, Y_{12}, \cdots, Y_{1n_1}, Y_{21}, \cdots, Y_{2n_2}, \cdots, Y_{I1}, \cdots, Y_{In_I}\right)^\top$ denote the n-dimensional random vector of response. Let $Y_{\cdot j}$ denote the jth component of Y. In other words, if we ignore the class information, then $Y_{\cdot j}$ represents the response (or the class membership) for the jth subject in the sample, where $j = 1, \cdots, n$.

For $i = 1, \cdots, I$, let $X_{li} = \left(X_{li1}, \cdots, X_{lin_i}\right)^\top$ denote the lth predictor (or covariate) vector associated with class i, where $l = 1, \cdots, p$ for a positive integer p. We write $X_l = \left(X_{l1}^\top, \cdots, X_{lI}^\top\right)^\top$ for $l = 1, \cdots, p$, and let $X = \left(X_1, \cdots, X_p\right)$ denote the $n \times p$ matrix of

predictors. Let $X_{\cdot j} = \left(X_{\cdot j1}, \cdots, X_{\cdot jp}\right)^\top$ denote the jth row of X, which represents the p-dimensional predictor vector for the jth subject. Without loss of generality, the $\left\{X_{\cdot j}, Y_{\cdot j}\right\}$ are treated as independent and identically distributed (i.i.d.) for $j = 1, \cdots, n$. We let lower case letters represent realized values for the corresponding random variables. For example, $x_{\cdot j}$ stands for a realized value of $X_{\cdot j}$. The data structure is shown in Table 1.

The objective here is to use the observed data to build models in order to predict the class label for a new subject using his/her observed predictor measurement.

Logistic regression model for multiclass response

With the multiclass response, we may consider the use of the logistic regression model by adapting the discussion of Agresti (2012, Section 7.1). For $i = 1, \cdots, I$ and $j = 1, \cdots, n$, let $\pi_{ij}(x_{\cdot j}) = P\left(Y_{\cdot j} = i | X_{\cdot j} = x_{\cdot j}\right)$ denote the conditional probability that subject j is selected from class i, given the predictor information $X_{\cdot j} = x_{\cdot j}$.

Noting the constraint $\sum_{i=1}^{I} \pi_{ij}(x_{\cdot j}) = 1$ for every $j = 1, \cdots, n$, to describe the $\pi_{ij}(x_{\cdot j})$, we can only model $(I-1)$ of the $\pi_{ij}(x_{\cdot j})$ rather than all of the $\pi_{ij}(x_{\cdot j})$. Without loss of generality, we take the Ith conditional probability $\pi_{Ij}(x_{\cdot j})$ as the reference and then consider the logistic model

$$\log\left\{\frac{\pi_{ij}(x_{\cdot j})}{\pi_{Ij}(x_{\cdot j})}\right\} = \gamma_{0i} + {\gamma_i}^\top x_{\cdot j} \tag{1}$$

for $i = 1, \cdots, I-1$ and $j = 1, \cdots, n$, where $\gamma = \left(\gamma_{01}, {\gamma_1}^\top, \gamma_{02}, {\gamma_2}^\top, \cdots, \gamma_{0,I-1}, {\gamma_{I-1}}^\top\right)^\top$ is the vector of parameters with the intercepts γ_{0i} and a p-dimensional vector γ_i of parameters.

Equivalently, (1) shows that for $i = 1, \cdots, I-1$ and $j = 1, \cdots, n$,

$$\pi_{ij}(x_{\cdot j}) = \frac{\exp\left(\gamma_{0i} + {\gamma_i}^\top x_{\cdot j}\right)}{1 + \sum_{l=1}^{I-1} \exp\left(\gamma_{0l} + {\gamma_l}^\top x_{\cdot j}\right)} \tag{2}$$

and

$$\pi_{Ij}(x_{\cdot j}) = 1 - \sum_{i=1}^{I-1} \pi_{ij}(x_{\cdot j}). \tag{3}$$

Since the distribution of the Y_{ij} can be delineated by a multinominal distribution, the likelihood function for the observed data is given by

$$L(\gamma) = \prod_{i=1}^{I} \left\{ \prod_{j=1}^{n} \pi_{ij}(x_{\cdot j})^{y_{ij}} \right\}, \tag{4}$$

where $\pi_{ij}(x_{\cdot j})$ is determined by (2) or (3). Estimation of γ can proceed with maximizing (4). Let $\widehat{\gamma} = \left(\widehat{\gamma}_{01}, \widehat{\gamma}_1^\top, \widehat{\gamma}_{02}, \widehat{\gamma}_2^\top, \cdots, \widehat{\gamma}_{0,I-1}, \widehat{\gamma}_{I-1}^\top\right)^\top$ denote the resulting maximum likelihood estimate of γ.

To predict the class label for a new subject with a p-dimensional predictor vector $\widetilde{x}$, we first calculate the right-hand side of (2) and (3) with the $\left(\gamma_{0i}, {\gamma_i}^\top\right)^\top$ replaced by the corresponding estimate obtained for the training data and let $\widehat{\pi}_1, \cdots, \widehat{\pi}_I$ denote the corresponding values. Let i^* denote the index which corresponds to the largest value of $\{\widehat{\pi}_1, \cdots, \widehat{\pi}_I\}$. Then the class label for this new subject is predicted as i^*.

Table 1 Two ways to display data with a multiclass response and predictors

Data Displayed With Class Label Used							Without Distinguishing Class Label				
Class	Subject	Predictor				Response	Subject	Predictor			Response
1	1	X_{111}	X_{211}	$\cdots$	X_{p11}	$Y_{11}=1$	1	$X_{\cdot 11}$	$\cdots$	$X_{\cdot 1p}$	$Y_{\cdot 1}$
	2	X_{112}	X_{212}	$\cdots$	X_{p12}	$Y_{12}=1$	2	$X_{\cdot 21}$	$\cdots$	$X_{\cdot 2p}$	$Y_{\cdot 1}$
	3	X_{113}	X_{213}	$\cdots$	X_{p13}	$Y_{13}=1$	3	$X_{\cdot 31}$	$\cdots$	$X_{\cdot 3p}$	$Y_{\cdot 3}$
	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$
	n_1	X_{11n_1}	X_{21n_1}	$\cdots$	X_{p1n_1}	$Y_{1n_1}=1$	n_1	$X_{\cdot n_1 1}$	$\cdots$	$X_{\cdot n_1 p}$	$Y_{\cdot n_1}$
2	1	X_{121}	X_{221}	$\cdots$	X_{p21}	$Y_{21}=2$	n_1+1	$X_{\cdot,n_1+1,1}$	$\cdots$	$X_{\cdot,n_1+1,p}$	$Y_{\cdot,n_1+1}$
	2	X_{122}	X_{222}	$\cdots$	X_{p22}	$Y_{22}=2$	n_1+2	$X_{\cdot,n_1+2,1}$	$\cdots$	$X_{\cdot,n_1+2,p}$	$Y_{\cdot,n_1+2}$
	3	X_{123}	X_{223}	$\cdots$	X_{p23}	$Y_{23}=2$	n_1+3	$X_{\cdot,n_1+3,1}$	$\cdots$	$X_{\cdot,n_1+3,p}$	$Y_{\cdot,n_1+3}$
	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$
	n_2	X_{12n_2}	X_{22n_2}	$\cdots$	X_{p2n_2}	$Y_{2n_2}=2$	n_1+n_2	$X_{\cdot,n_1+n_2,1}$	$\cdots$	$X_{\cdot,n_1+n_2,p}$	$Y_{\cdot,n_1+n_2}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$
l	1	X_{1l1}	X_{2l1}	$\cdots$	X_{pl1}	$Y_{l1}=l$	$n-n_l+1$	$X_{\cdot,n-n_l+1,1}$	$\cdots$	$X_{\cdot,n-n_l+1,p}$	$Y_{\cdot,n-n_l+1}$
	2	X_{1l2}	X_{2l2}	$\cdots$	X_{pl2}	$Y_{l2}=l$	$n-n_l+2$	$X_{\cdot,n-n_l+2,1}$	$\cdots$	$X_{\cdot,n-n_l+2,p}$	$Y_{\cdot,n-n_l+2}$
	3	X_{1l3}	X_{2l3}	$\cdots$	X_{pl3}	$Y_{l3}=l$	$n-n_l+3$	$X_{\cdot,n-n_l+3,1}$	$\cdots$	$X_{\cdot,n-n_l+3,p}$	$Y_{\cdot,n-n_l+3}$
	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$
	n_l	X_{1ln_l}	X_{2ln_l}	$\cdots$	X_{pln_l}	$Y_{ln_l}=l$	n	$X_{\cdot,n,1}$	$\cdots$	$X_{\cdot,n,p}$	$Y_{\cdot,n}$

Classification with predictor graphical structures accommodated

In this section, we propose two classification methods for prediction which incorporate the network structure of the predictors. We first describe the use of graphical models to facilitate the association structure of the predictors, and then explore two methods of building prediction models using the identified association structures.

Predictor network structure

Graphical models are useful to facilitate the network structures of the predictors. Here we describe the way of using graphical models to delineate possible association structures of the predictors. For $j = 1, \cdots, n$, we use an undirected *graph*, denoted as $G_j = (V_j, E_j)$, to describe the relationship among the components of $X_{\cdot j} = \left(X_{\cdot j1}, \cdots, X_{\cdot jp}\right)^\top$, where $V_j = \left\{1, \cdots, p\right\}$ includes all the indices of predictors and $V_j \times V_j$ contains all pairs with unequal coordinates. A covariate $X_{\cdot jr}$ is called a *vertex* of the graph G_j if $r \in V_j$; a pair of predictors $\left\{X_{\cdot jr}, X_{\cdot js}\right\}$ is called an *edge* of the graph G_j if $(r, s) \in E_j \subset V_j \times V_j$. In the setting we consider, the sets V_j and E_j are common for $j = 1, \cdots, n$, so we let V and E denote the vertex and edge of the graph, respectively.

To characterize the distribution of the predictor $X_{\cdot j}$, we consider the graphical model with the exponential family distribution,

$$f(x_{\cdot j}; \beta, \Theta) = \exp\left\{\sum_{r \in V} \beta_r B(x_{\cdot jr}) + \sum_{(s,t) \in E} \theta_{st} B(x_{\cdot js}) B(x_{\cdot jt}) + \sum_{r \in V} C(x_{\cdot jr}) - A(\beta, \Theta)\right\}, \quad (5)$$

where $\beta = \left(\beta_1, \cdots \beta_p\right)^\top$ is a p-dimensional vector of parameters, $\Theta = [\theta_{st}]$ is a $p \times p$ symmetric matrix with zero diagonal elements, and $B(\cdot)$ and $C(\cdot)$ are given functions. The function $A(\beta, \Theta)$ is the normalizing constant which makes (5) integrated as 1; this function is also called the *log-partition function*, given by

$$A(\beta, \Theta) = \log \int \exp\left\{\sum_{r \in V} \beta_r B(x_{\cdot jr}) + \sum_{(s,t) \in E} \theta_{st} B(x_{\cdot js}) B(x_{\cdot jt}) + \sum_{r \in V} C(x_{\cdot jr})\right\} dx_{\cdot j}.$$

Formulation (5) gives a broad class of models which essentially covers most commonly used distributions. For example, if $B(x) = \frac{x}{\sigma}$ and $C(x) = -\frac{x^2}{2\sigma^2}$ where σ is a positive constant, then (5) yields the well-known *Gaussian graphical model* (Friedman et al. 2008; Hastie et al. 2015; Lee and Hastie 2015). If $B(x) = x$ and $C(x) = 0$ with $x \in \{0, 1\}$, then with the β_r set to be zero, (5) reduces to

$$\exp\left\{\sum_{(s,t) \in E} \theta_{st} x_{\cdot js} x_{\cdot jt} - A(\Theta)\right\}, \quad (6)$$

which is the *Ising model* without the singletons (Ravikumar et al. 2010).

To focus on featuring the pairwise association among the components of $X_{\cdot j}$, similar to the structure of (6), we consider the following graphical model

$$f(x_{\cdot j};\Theta)=\exp\left\{\sum_{(s,t)\in E}\theta_{st}x_{\cdot js}x_{\cdot jt}+\sum_{r\in V}C(x_{\cdot jr})-A(\Theta)\right\}, \tag{7}$$

where the function $A(\Theta)$ is the normalizing constant, and the θ_{st} and $C(\cdot)$ are defined as for (5). Model (7) is a special case of (5) which constraints the main effects parameters β_r in (5) to be zero; nonzero parameter θ_{st} implies that $X_{\cdot js}$ and $X_{\cdot jt}$ are conditionally dependent given other predictors.

To estimate Θ, one may apply the likelihood method using the distribution (7) directly. Alternatively, a simpler estimation method can be carried out based on a conditional distribution derived from (7) (Meinshausen and Bühlmann2006; Hastie et al. 2015, p.254). For every $s\in V$, let $X_{\cdot j,V\setminus\{s\}}$ denote the $(p-1)$-dimensional subvector of $X_{\cdot j}$ with its sth component deleted, i.e., $X_{\cdot j,V\setminus\{s\}}=\left(X_{\cdot j1},\cdots,X_{\cdot j,s-1},X_{\cdot j,s+1},\cdots,X_{\cdot jp}\right)^{\top}$. By some algebra, we have

$$f\left(x_{\cdot js}|x_{\cdot j,V\setminus\{s\}};\theta_s\right)=\exp\left\{x_{\cdot js}\left(\sum_{t\in V\setminus\{s\}}\theta_{st}x_{\cdot jt}\right)+C\left(x_{\cdot js}\right)-D\left(\sum_{t\in V\setminus\{s\}}\theta_{st}x_{\cdot jt}\right)\right\}, \tag{8}$$

where $D(\cdot)$ is the normalizing constant ensuring the integration of (8) equal one, and $\theta_s=\left(\theta_{s1},\cdots,\theta_{s,s-1},\theta_{s,s+1},\cdots,\theta_{sp}\right)^{\top}$ is a $(p-1)$-dimensional vector of parameters indicating the relationship of $X_{\cdot js}$ with all other predictors $X_{\cdot jr}$ for $r\in\{1,\cdots,p\}\setminus\{s\}$ associated with (8).

Let $\ell(\theta_s)$ be the log-likelihood for θ_s multiplied with $-\frac{1}{n}$ with the constand omitted, i.e.,

$$\begin{aligned}\ell(\theta_s)&=-\frac{1}{n}\log\left\{\prod_{j=1}^{n}f\left(x_{\cdot js}|x_{\cdot j,V\setminus\{s\}};\theta_s\right)\right\}\\&=\frac{1}{n}\sum_{j=1}^{n}\left\{-x_{\cdot js}\left(\sum_{t\in V\setminus\{s\}}\theta_{st}x_{\cdot jt}\right)+D\left(\sum_{t\in V\setminus\{s\}}\theta_{st}x_{\cdot jt}\right)\right\}.\end{aligned}$$

Then an estimator of θ_s can be obtained as

$$\widehat{\theta}_s(\lambda)=\underset{\theta_s}{\operatorname{argmin}}\left\{\ell(\theta_s)+\lambda\|\theta_s\|_1\right\}, \tag{9}$$

where λ is a tuning parameter and $\|\cdot\|_1$ is the L_1-norm. In principle, the L_1-norm in (9) may be replaced by other penalty functions such as the weighted L_1-norm (Zou 2006) and the nonconcave function (Fan and Li 2001). Here we focus on using the L_1-norm, the well-known LASSO penalty (Tibshirani 1996), to determine informative pairwise dependent predictors. The LASSO penalty is frequently considered when dealing with graphical models; it has been implemented in R. For instance, R packages `huge` and `XMRF` use the LASSO penalty to determine the network structure.

We comment that the estimator obtained from (9) depends on the choice of the tuning parameter λ. There is no unique way of selecting a suitable tuning parameter, and methods such as the Akaike information criterion (AIC), the Bayesian information criterion (BIC), the Cross Validation (CV), and the Generalized Cross Validation (GCV) may be considered in the selection of the tuning parameter. Suggested by Wang et al. (2007), BIC tends to outperform others in many situations, especially in the setting with a penalized likelihood function. Consequently, here we employ the BIC approach to select the tuning parameter λ.

Define

$$BIC(\lambda) = 2n\ell\left(\widehat{\theta}_s(\lambda)\right) + \log(n) \times \mathrm{df}\left\{\widehat{\theta}_s(\lambda)\right\}, \tag{10}$$

where $\mathrm{df}\left\{\widehat{\theta}_s(\lambda)\right\}$ represents the number of non-zero elements in $\widehat{\theta}_s(\lambda)$ for a given λ. The optimal tuning parameter λ, denoted by $\widehat{\lambda}$, is determined by minimizing (10) within a suitable range of λ. As a result, the estimator of θ_s is determined by $\widehat{\theta}_s = \widehat{\theta}_s\left(\widehat{\lambda}\right)$.

The preceding procedure is repeated for all $s \in V$ and yields the estimator $\widehat{\theta}_s$ for all $s \in V$. There is an important point we need to pay attention. For $(s,t) \in E$, the estimates $\widehat{\theta}_{st}$ and $\widehat{\theta}_{ts}$ are not necessarily identical although θ_{st} and θ_{ts} are constrained to be equal. To overcome this problem, we apply the AND rule (Meinshausen and Bühlmann 2006; Hastie et al. 2015, p.255) to determine the final estimates of $\widehat{\theta}_{st}$ and $\widehat{\theta}_{ts}$ as their maximum if both $\widehat{\theta}_{st}$ and $\widehat{\theta}_{ts}$ are not zero; and set $\widehat{\theta}_{st}$ and $\widehat{\theta}_{ts}$ to be zero if one of them is zero.

To determine an estimated set of edges, we define

$$\widehat{\mathcal{N}}(s) = \left\{t \in V : \widehat{\theta}_{st} \neq 0\right\}$$

for $s \in V$. Then

$$\widehat{E} = \left\{(s,t) : s \in \widehat{\mathcal{N}}(t) \text{ and } t \in \widehat{\mathcal{N}}(s)\right\} \tag{11}$$

is taken as the set of the edges that are estimated to exist. The R package '`huge`' can be implemented to show the graphic results.

Under mild regularity conditions, the estimated set of edges $\widehat{E}$ approximate the true network structure E accurately, as shown below which was available in Ravikumar et al. (2010, Section 2.2) and Theorem 5 (b) of Yang et al. (2015).

Proposition 1 (Network Recovery) *Suppose E is the set of edges, and let $\widehat{E}$ be the estimated set of edges. Under regular conditions in Meinshausen and Bühlmann (2006), we have that as $n \to \infty$,*

$$P\left(\widehat{E} = E\right) \to 1.$$

Logistic regression with homogeneous graphically structured predictors

To incorporate the network structures of the predictors into building a prediction model, in the next two subsections, we present two methods which can be readily implemented using the R package `huge` and the R function `glm` for fitting a logistic regression model.

In the first method, called the *logistic regression with homogeneous graphically structured predictors* (LR-HomoGraph) method, we consider the case where the subjects in different classes share a common network structure in the predictors. To build a prediction model, we make use of the development of the logistic model with multiclass responses, discussed by Agresti (2007, Section 6.1) and Agresti (2012, Section 7.1).

We first identify the pairwise dependence of the predictors using the measurements of all the subjects without distinguishing their class labels. Let $\widehat{\theta}_{st}$ be the estimate for θ_{st} obtained for (9) by using all the predictor measurements of $\left\{X_{\cdot j} : j = 1, \cdots, n\right\}$, and let $\widehat{E} = \left\{(s,t) : \widehat{\theta}_{st} \neq 0\right\}$ denote the resulting estimated set of edges.

Next, for $i = 1, \cdots, I$ and $j = 1, \cdots, n$, we let

$$p_{ij}(x_{\cdot j}) = P\left(Y_{\cdot j} = i \middle| X_{\cdot j} = x_{\cdot j}\right)$$

be the conditional probability of $Y_{\cdot j} = i$ given $X_{\cdot j} = x_{\cdot j}$. Consider the logistic regression model

$$p_{ij}(x_{\cdot j}) = \frac{\exp\left(\alpha_{i0} + \sum\limits_{(s,t)\in\widehat{E}} \alpha_{i,st} x_{\cdot js} x_{\cdot jt}\right)}{1 + \sum\limits_{l=1}^{I-1} \exp\left(\alpha_{l0} + \sum\limits_{(s,t)\in\widehat{E}} \alpha_{l,st} x_{\cdot js} x_{\cdot jt}\right)} \tag{12}$$

for $i = 1, 2, \cdots, I-1$, where $(\alpha_{i0}, \alpha_{i,st})^\top$ is the vector of parameters associated with class i and the constraint $\sum\limits_{i=1}^{I} p_{ij}(x) = 1$ is imposed for every $j = 1, \cdots, n$.

For subject $j = 1, \cdots, n$, we let $Y^*_{ij} = 1$ if subject j is in class i and $Y^*_{ij} = 0$ otherwise, and hence, $\sum\limits_{i=1}^{I} Y^*_{ij} = 1$ for every j. Let y^*_{ij} denote a realized value of Y^*_{ij}. For $i = 1, \cdots, I$ and $j = 1, \cdots, n$, the likelihood function is given by (Agresti 2012, p.273)

$$L(\alpha) = \prod_{i=1}^{I} \left\{ \prod_{j=1}^{n} p_{ij}(x_{\cdot j})^{y^*_{ij}} \right\}, \tag{13}$$

where $\alpha = \left(\alpha_{10}, \alpha_{1\cdot}^\top, \cdots, \alpha_{(I-1)0}, \alpha_{(I-1)\cdot}^\top\right)^\top$ is the vector of parameters with vector $\alpha_{i\cdot} = \left(\alpha_{i,st} : (s,t) \in \widehat{E}\right)^\top$ for $i = 1, \cdots, I-1$.

The estimator $\widehat{\alpha}$ can be derived by maximizing (13) with respect to α. Therefore, for the realization $x_{\cdot j}$ of the p-dimensional vector $X_{\cdot j}$, $p_{ij}(x_{\cdot j})$ is estimated as

$$\widehat{p}_{ij}(x_{\cdot j}) = \frac{\exp\left(\widehat{\alpha}_{i0} + \sum\limits_{(s,t)\in\widehat{E}} \widehat{\alpha}_{i,st} x_{\cdot js} x_{\cdot jt}\right)}{1 + \sum\limits_{l=1}^{I-1} \exp\left(\widehat{\alpha}_{l0} + \sum\limits_{(s,t)\in\widehat{E}} \widehat{\alpha}_{l,st} x_{\cdot js} x_{\cdot jt}\right)} \quad \text{for } i = 1, \cdots, I-1, \tag{14}$$

and $p_{Ij}(x_{\cdot j})$ is estimated as

$$\widehat{p}_{Ij}(x_{\cdot j}) = 1 - \sum_{i=1}^{I-1} \widehat{p}_{ij}(x_{\cdot j}). \tag{15}$$

Finally, to predict the class label for a new subject with a p-dimensional predictor $\widetilde{x}$, we first calculate the right-hand side of (14) and (15), and let $\widetilde{\widehat{p}}_1, \cdots, \widetilde{\widehat{p}}_I$ denote the corresponding values. Let i^* denote the index which corresponds to the largest value of $\left\{\widetilde{\widehat{p}}_1, \cdots, \widetilde{\widehat{p}}_I\right\}$, i.e., $i^* = \underset{1\le i\le I}{\text{argmax}} \widetilde{\widehat{p}}_i$. Then the class label for this new subject is predicted as i^*.

Logistic regression with class-dependent graphically structured predictors

We now present an alternative to the method described in "Logistic regression with homogeneous graphically structured predictors" section. Instead of pooling all the covariates to feature the covariate network structure, this method, called the *logistic regression with class-dependent graphically structured covariates* (LR-ClassGraph) method, stratifies the covariate information by class when characterizing the covariate network structures.

We first introduce a binary, surrogate response variable Y_{ij}^i for every i and j, where $i = 1, \cdots, I$ and $j = 1, \cdots, n$. Let

$$Y_{ij}^i = \begin{cases} 1, & Y_{ij} = i, \\ 0, & \text{otherwise}, \end{cases}$$

and define $Y^i = \left(0, \cdots, 0, Y_{i1}^i, \cdots, Y_{in_i}^i, 0, \cdots, 0\right)^\top$ to be an n-dimensional vector whose elements corresponding to class i are respectively $Y_{i1}^i, \cdots, Y_{in_i}^i$, and other elements are zero. That is, $Y^i = (\underbrace{0, \cdots, 0}_{n_1+\cdots+n_{i-1}}, \underbrace{1, \cdots, 1}_{n_i}, \underbrace{0, \cdots, 0}_{n_{i+1}+\cdots+n_I})^\top$ with $i = 1, \cdots, I$. Now we implement the following steps.

Step 1: (Class-Dependent Predictor Network)

For each class $i = 1, \cdots, I$, we apply the procedure described in "Predictor network structure" section to determine the network structure of predictors in class i. Let $\widehat{E}^i = \left\{(s,t) : \widehat{\theta}_{st}^i \neq 0\right\}$ denote an estimated set of edges for class i, where $\widehat{\theta}_{st}^i$ is the estimate of θ_{st} derived from (9) based on using the predictor measurements in class i.

Step 2: (Class-Dependent Model Building)

For each class $i = 1, \cdots, I$, fit a logistic regression model using the surrogate response vector Y^i with the estimated covariates network structure $\widehat{E}^i$ incorporated. Specifically, for the jth component of Y^i, Y_j^i, define $\pi_j^i(x_{\cdot j}) = P\left(Y_j^i = 1 | X_{\cdot j} = x_{\cdot j}\right)$ and consider the logistic regression model

$$\text{logit}\left\{\pi_j^i(x_{\cdot j})\right\} = \gamma_0^i + \sum_{(s,t)\in\widehat{E}^i} \gamma_{st}^i x_{\cdot js} x_{\cdot jt}, \tag{16}$$

where $j = 1, \cdots, n$, $\left(\gamma_0^i, \gamma_{st}^i\right)^\top$ is the vector of parameters associated with class i. By the theory of maximum likelihood (e.g., Agresti 2012), we obtain the estimate $\left(\widehat{\gamma}_0^i, \widehat{\gamma}_{st}^i\right)^\top$ of $\left(\gamma_0^i, \gamma_{st}^i\right)^\top$.

Step 3: (Prediction)

For a realization $x_{\cdot j}$ of the p-dimensional vector $X_{\cdot j}$, based on (16), $\pi_j^i(x_{\cdot j})$ can be estimated by

$$\widehat{\pi}_j^i(x_{\cdot j}) = \frac{\exp\left(\widehat{\gamma}_0^i + \sum_{(s,t)\in\widehat{E}^i} \widehat{\gamma}_{st}^i x_{\cdot js} x_{\cdot jt}\right)}{1 + \exp\left(\widehat{\gamma}_0^i + \sum_{(s,t)\in\widehat{E}^i} \widehat{\gamma}_{st}^i x_{\cdot js} x_{\cdot jt}\right)} \quad \text{for } i = 1, \cdots, I. \tag{17}$$

To predict the class label for a new subject with a p-dimensional covariate vector $\widetilde{x}$, we first calculate (17) with $x_{\cdot j}$ replaced by $\widetilde{x}$ for $i = 1, \cdots, I$, and let $\widetilde{\widehat{\pi}^1}, \cdots, \widetilde{\widehat{\pi}^I}$ denote the corresponding values. Let i^* denote the index which corresponds to the largest value of $\left\{\widetilde{\widehat{\pi}^1}, \cdots, \widetilde{\widehat{\pi}^I}\right\}$, i.e.,

$$\widetilde{\widehat{\pi}^{i^*}} = \max_{1\le i\le I} \widetilde{\widehat{\pi}^i}. \tag{18}$$

Then the class label for this new subject is predicted as i^*.

Comparison of decision boundaries

As noted in "Logistic regression with homogeneous graphically structured predictors" and "Logistic regression with class-dependent graphically structured predictors" sections, while both the LR-HomoGraph and LR-ClassGraph methods employ logistic regression to classify classes, they are different in the way of featuring predictor structures. Furthermore, we may compare their differences in terms of decision boundaries.

First, we examine the decision boundaries for the LR-HomoGraph method. For $i \neq k$, the boundary between the ith and kth classes is determined by

$$\widehat{p}_{ij}(x_{\cdot j}) = \widehat{p}_{ik}(x_{\cdot j})$$

for a new instance with the predictor value $x_{\cdot j}$, where $\widehat{p}_{ij}(x_{\cdot j})$ and $\widehat{p}_{ik}(x_{\cdot j})$ are given by (14) or (15). To be more specific, for any $i = 1, ..., I-1$, if $k = 1, ..., I-1$ and $k \neq i$, then by (14), the boundary between the ith and kth classes is

$$\sum_{(s,t)\in\widehat{E}} (\widehat{\alpha}_{i,st} - \widehat{\alpha}_{k,st}) x_{\cdot js} x_{\cdot jt} + (\widehat{\alpha}_{i0} - \widehat{\alpha}_{k0}) = 0; \tag{19}$$

and the boundary between the ith and Ith classes is, by (15),

$$\sum_{(s,t)\in\widehat{E}} \widehat{\alpha}_{i,st} x_{\cdot js} x_{\cdot jt} + \widehat{\alpha}_{i0} = 0. \tag{20}$$

Similarly, the decision boundaries for the LR-ClassGraph method can be determined based on (17). For $i \neq k$, equating $\widehat{\pi}_j^i(x_{\cdot j})$ and $\widehat{\pi}_j^k(x_{\cdot j})$ for a covariate value $x_{\cdot j}$ gives the boundary between the ith and kth classes

$$\sum_{(s,t)\in\widehat{E}^i} \widehat{\gamma}_{st}^i x_{\cdot js} x_{\cdot jt} - \sum_{(s,t)\in\widehat{E}^k} \widehat{\gamma}_{st}^k x_{\cdot js} x_{\cdot jt} + \left(\widehat{\gamma}_0^i - \widehat{\gamma}_0^k\right) = 0. \tag{21}$$

Comparing (21) to (19) or (20) shows that decision boundaries for both the LR-HomoGraph and LR-ClassGraph methods are all quadratic surfaces determined by the features selected from the graphical models. However, the way of incorporating the features is different for the two methods. The boundaries (21) are determined by the quadratic terms identified using instances from classes i and k separately, but the quadratic terms in the boundary (19) or (20) are not distinguished by the class labels. In addition, the coefficients $\widehat{\gamma}_{st}^i$ and $\widehat{\alpha}_{i,st}$ associated with the decision boundaries are generally different.

Evaluation of the performance

In this section we discuss the evaluation of the procedures proposed in "Logistic regression with homogeneous graphically structured predictors" and "Logistic regression with class-dependent graphically structured predictors" sections. For comparisons, we also examine some conventional classification methods in machine learning, including support vector machine (SVM), linear discriminant analysis (LDA), K-nearest neighbor (KNN), and extreme gradient boosting (XGBOOST). We first describe the measures of assessing the prediction error that are commonly used, and then we briefly review the four classification methods.

Criteria for performances

In this subsection, we describe several criteria of evaluating the performance for prediction. To show the overall performance of prediction, we consider either *micro averaged*

metrics or *macro averaged metrics* (Parambath et al. 2018). For subject $j = 1, \cdots, n$, let $\widehat{y}_{\cdot j}$ denote the predicted class label. For class $i = 1, \cdots, I$, we calculate the number of the *true positives*, the number of the *false positives*, and the number of the *false negatives*, respectively, given by

$$\mathrm{TP}_i = \sum_{j=1}^{n} \mathbb{I}\left(y_{\cdot j} = i, \widehat{y}_{\cdot j} = i\right), \mathrm{FP}_i = \sum_{j=1}^{n} \mathbb{I}\left(y_{\cdot j} \neq i, \widehat{y}_{\cdot j} = i\right),$$

and

$$\mathrm{FN}_i = \sum_{j=1}^{n} \mathbb{I}\left(y_{\cdot j} = i, \widehat{y}_{\cdot j} \neq i\right),$$

where $\mathbb{I}(\cdot)$ is the indicator function. For *micro* averaged metrics, we define *precision* and *recall*, respectively, given by

$$PRE_{micro} = \frac{\sum_{i=1}^{I} \mathrm{TP}_i}{\sum_{i=1}^{I} \mathrm{TP}_i + \sum_{i=1}^{I} \mathrm{FP}_i} \text{ and } REC_{micro} = \frac{\sum_{i=1}^{I} \mathrm{TP}_i}{\sum_{i=1}^{I} \mathrm{TP}_i + \sum_{i=1}^{I} \mathrm{FN}_i}.$$

Then *Micro-F-score* is defined as

$$F_{micro} = 2 \times \frac{PRE_{micro} \times REC_{micro}}{PRE_{micro} + REC_{micro}}. \tag{22}$$

On the other hand, for *macro* averaged metrics, for $i = 1, \cdots, I$, let $PRE_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FP}_i}$ denote *precision* for class i, and let $REC_i = \frac{\mathrm{TP}_i}{\mathrm{TP}_i + \mathrm{FN}_i}$ denote *recall* for class i. Then the overall *precision* and *recall* are, respectively, defined as

$$PRE_{macro} = \frac{1}{I} \sum_{i=1}^{I} PRE_i \text{ and } REC_{macro} = \frac{1}{I} \sum_{i=1}^{I} REC_i;$$

and *Macro-F-score* is defined as

$$F_{macro} = 2 \times \frac{PRE_{macro} \times REC_{macro}}{PRE_{macro} + REC_{macro}}. \tag{23}$$

In principle, higher values of *PRE*, *REC* and *F* based on both micro and macro reflect better performance of methods (Parambath et al. 2018; Sokolova et al. 2006).

Support vector machine for multiclass responses

Support vector machine (SVM) was originally designed for two-class classification (Hastie et al. 2008, Sec. 12.2), and its extensions to the multiclass responses have been discussed by many authors. An early extension of the SVM to accommodating multiclass classification is the *one-against-all* method (Hsu and Lin 2002). The main idea is that the ith SVM is trained from all subjects with positive labels in the ith class and all other subjects with negative labels. This type of SVM for multiclass classification, however, ignores the heterogeneity among the subjects in each class.

A useful multiclass SVM is the *one-against-one* method (Knerr et al. 1990), which is implemented in the R package `e1071`. Different from the one-against-all method, the one-against-one method first produces $I(I-1)/2$ pairwise classifiers and trains data from any two selected classes, and then it applies SVM with binary classification to each pairwise classifiers. To see this, for $i_1, i_2 \in \{1, \cdots, I\}$ with $i_1 < i_2$, we consider the following optimization

$$\min_{w^{i_1i_2},b^{i_1i_2},\xi_j^{i_1i_2}} \left\{ \frac{1}{2}\left(w^{i_1i_2}\right)^\top w^{i_1i_2} + C\sum_{j=1}^{n}\xi_j^{i_1i_2} \right\} \tag{24}$$

subject to

$$\begin{aligned} &\text{for } j = 1,\ldots,n, \\ &\quad \xi_j^{i_1i_2} \geq 0 \\ &\quad \left\{\left(w^{i_1i_2}\right)^\top \phi(X_{\cdot j}) + b^{i_1i_2}\right\} \geq 1 - \xi_j^{i_1i_2}, \text{if} Y_{\cdot j} = i_1, \\ &\quad \left\{\left(w^{i_1i_2}\right)^\top \phi(X_{\cdot j}) + b^{i_1i_2}\right\} \leq -1 + \xi_j^{i_1i_2}, \text{if} Y_{\cdot j} = i_2, \end{aligned}$$

where $\phi(\cdot)$ is a non-linear mapping from a p-dimensional vector to a q-dimensional vector with $q > p$ (Hsu and Lin 2002), $w^{i_1i_2}$ is a q-dimensional vector of parameters associated with the comparison between classes i_1 and i_2, $b^{i_1i_2}$ is a scalar, $\xi_j^{i_1i_2}$ is the slack variable for the soft margin solution, and C is a cost parameter controlling balance of maximizing the margin and minimizing the training error.

Solving (24) for arbitrary $i_1, i_2 \in \{1,\cdots,I\}$ with $i_1 < i_2$ yields $I(I-1)/2$ classifiers and those classifiers can then be used for classification of a new instance, say $\widetilde{X} = \widetilde{x}$. This can be done through a voting process (Hsu and Lin 2002). Specifically, let $\mathcal{L} = \{(1,2),(1,3),\cdots,(1,I),\ (2,3),\cdots,(2,I),\cdots,(I-1,I)\}$ be the collection of all pairwise class labels which includes $I(I-1)/2$ elements. For each class i with $i = 1,\cdots,I$, we let $vote(i)$ denote the "number of vote" related to class i. Then we carry out the following three steps.

Step 1: For class $i = 1,\cdots,I$, the initial value of $vote(i)$ is set as 0.

Step 2: For any given class i, we consider a subcollection of $\mathcal{L}$, $\left\{(i,i') : i' = i+1,\cdots,I\right\}$, which is associated with class i. Calculate sign $\left\{(w^{ii'})^\top\phi(\widetilde{x}) + b^{ii'}\right\}$ *repeatedly* for $i' = i+1,\cdots,I$ and then determine the values of $vote(i)$ and $vote(i')$ *iteratively* by the rule:

$$\begin{aligned} &\text{If sign}\left\{(w^{ii'})^\top\phi(\widetilde{x}) + b^{ii'}\right\} > 0, \text{ then we let} \\ &\qquad vote(i) = vote(i) + 1; \\ &\text{otherwise,} \\ &\qquad vote(i') = vote(i') + 1; \end{aligned}$$

where $vote(i')$ on the right-hand-side of the equation is a value determined by the previous step, $vote(i')$ on the left-hand-side of the equation represents a newly determined value, and $i' = i+1,\cdots,I$.

Step 3: Repeat Step 2 for $i = 1,\cdots,I$. In this way, we determine all the final values of $vote(1),\cdots,vote(I)$. Let i^* denote the class index corresponding to the largest value of $\{vote(1),\cdots,vote(I)\}$, i.e., $i^* = \underset{1\leq i\leq I}{\text{argmax}}\{vote(i)\}$. Then we let i^* be the predicted class for the new instance.

Linear discriminant analysis

The idea of LDA is to model the distribution of the predictors $X_{\cdot j}$ separately for each of the classes $Y_{\cdot j}$, and then use the Bayes theorem to obtain the conditional probabilities $P(Y_{\cdot j} = i | X_{\cdot j} = x_{\cdot j})$ (e.g., James et al. 2017). For $i = 1,\cdots,I$ and $j = 1,\cdots,n$, let $f_{j|i}(x_{\cdot j})$ denote the conditional probability density function of the predictor $X_{\cdot j}$ taking value $x_{\cdot j}$

given that subject j comes from the ith class. Let $\pi_{i,j} = P\left(Y_{\cdot j} = i\right)$ denote the probability that the jth subject is randomly selected from class i. It is immediate that $\sum_{i=1}^{I} \pi_{i,j} = 1$ for $j = 1, \cdots, n$. By some algebra (Hastie et al. 2008, p.108) and the Bayes theorem, we obtain the posterior probability

$$P\left(Y_{\cdot j} = i | X_{\cdot j} = x_{\cdot j}\right) = \frac{f_{j|i}(x_{\cdot j})\pi_{i,j}}{\sum_{l=1}^{I} f_{j|l}(x_{\cdot j})\pi_{l,j}} \tag{25}$$

for $i = 1, \cdots, I$ and $j = 1, \cdots, n$.

To compare two classes i and l with $i \neq l$, we calculate the log-ratio of (25) for classes i and l, given by

$$\log\left\{\frac{P\left(Y_{\cdot j} = i | X_{\cdot j} = x_{\cdot j}\right)}{P\left(Y_{\cdot j} = l | X_{\cdot j} = x_{\cdot j}\right)}\right\} = \log\left(\frac{f_{j|i}(x_{\cdot j})}{f_{j|l}(x_{\cdot j})}\right) + \log\left(\frac{\pi_{i,j}}{\pi_{l,j}}\right). \tag{26}$$

To elaborate on the idea, we particularly consider the case where the conditional distribution $f_{j|i}(x_{\cdot j})$ of $X_{\cdot j}$ given $Y_{\cdot j} = i$ is assumed to be the normal distribution $N(\mu_i, \Sigma_i)$ with the probability density function

$$f_{j|i}(x_{\cdot j}) = \frac{1}{(2\pi)^{p/2}|\Sigma_i|^{1/2}} \exp\left\{-\frac{1}{2}\left(x_{\cdot j} - \mu_i\right)^{\top} \Sigma_i^{-1}\left(x_{\cdot j} - \mu_i\right)\right\}. \tag{27}$$

If the covariance matrices Σ_i in (27) are assumed to be common, i.e., $\Sigma_i = \Sigma$ for every i where Σ is a positive definite matrix, (26) becomes

$$\log\left(\frac{\pi_{i,j}}{\pi_{l,j}}\right) - \frac{1}{2}\left(\mu_i + \mu_l\right)^{\top} \Sigma^{-1}\left(\mu_i + \mu_l\right) + x_{\cdot j}^{\top} \Sigma^{-1}\left(\mu_i + \mu_l\right). \tag{28}$$

If (28) > 0, then

$$P\left(Y_{\cdot j} = i | X_{\cdot j} = x_{\cdot j}\right) > P\left(Y_{\cdot j} = l | X_{\cdot j} = x_{\cdot j}\right),$$

showing that subject j with predictors $X_{\cdot j} = x_{\cdot j}$ is more likely to be selected from class i than from class l. Consequently, (28) defines a boundary between classes i and l which is a linear function of $x_{\cdot j}$.

Motivated by the form of (28), we consider a linear function in x

$$\delta_i(x) = \log\left(\pi_i\right) - \frac{1}{2}\mu_i^{\top} \Sigma^{-1} \mu_i + x^{\top} \Sigma^{-1} \mu_i, \tag{29}$$

where μ_i, π_i, and Σ are estimated by $\widehat{\mu}_i = \frac{1}{n_i} \sum_{y_{\cdot j} = i} x_{\cdot j}$, $\widehat{\pi}_i = \frac{n_i}{n}$, and $\widehat{\Sigma} = \frac{1}{n-I} \sum_{i=1}^{I} \sum_{y_{\cdot j} = i} \left(x_{\cdot j} - \widehat{\mu}_i\right)\left(x_{\cdot j} - \widehat{\mu}_i\right)^{\top}$, respectively. That is, (29) can be estimated by

$$\widehat{\delta}_i(x) = \log\left(\widehat{\pi}_i\right) - \frac{1}{2}\widehat{\mu}_i^{\top} \widehat{\Sigma}^{-1} \widehat{\mu}_i + x^{\top} \widehat{\Sigma}^{-1} \widehat{\mu}_i. \tag{30}$$

Function (30) is called the *linear discriminant function* and is used to determine the class label for a new instance (James et al. 2017, p.143; Hastie et al. 2008, p. 109). For the prediction of a new subject with covariate $\widetilde{x}$, we first calculate $\widehat{\delta}_i(\widetilde{x})$ using (30) for $i = 1, \cdots, I$. Next, we find i^* which is defined as

$$i^* = \underset{i=1,\cdots,I}{\operatorname{argmax}}\, \widehat{\delta}_i(\widetilde{x});$$

and the class label for this subject is then predicted as i^*.

K-nearest neighbor

The third classification method we compare with is the K-nearest neighbor (KNN) method which is a non-parametric approach. The key idea of KNN is to use the available instances to estimate the conditional probability of $Y_{\cdot j}$ given $X_{\cdot j}$, and then classify a new instance to a certain class based on the highest estimated conditional probability.

For a positive integer K and a new instance $\widetilde{x}$ of predictors $\widetilde{X}$, the first step of KNN is to identify K points which are closest to $\widetilde{x}$; let $\mathcal{N}_0(\widetilde{x})$ denote the set containing such K-nearest points of $\widetilde{x}$. Next, for $i = 1, \cdots, I$, we calculate

$$\widehat{\pi}_i = \frac{1}{K} \sum_{j' \in \mathcal{N}_0(\widetilde{x})} \mathbb{I}(y_{\cdot j'} = i).$$

Finally, let i^* denote the class label which corresponds to the largest value of $\{\widehat{\pi}_1, \cdots, \widehat{\pi}_I\}$. Then the class label for this new subject is predicted as i^*.

For the KNN method, a crucial issue is the selection of K. A small value of K usually yields an over-flexible decision boundary, which makes the classifier have a small bias but a large variance. On the contrary, with a large K, the boundary becomes less flexible and is close to linear, and classifier would have a small variance but a large bias. To determine an optimal K from the theoretical perspective, James et al. (2017, p. 184 and p. 186) suggested to use the cross-validation method to select K; but from the computational viewpoint, sometimes, a choice of K may be based on a random guess, as commented by James et al. (2017, p. 167).

Extreme gradient boosting

The extreme gradient boosting (XGBOOST) is a tree based ensemble method created under the gradient boosting framework (e.g., Chen and Guestrin 2016) and can be implemented by the R package `xgboost`.

Let $\mathcal{F}$ denote the space of functions representing regression trees f, where for $f \in \mathcal{F}$ with $f(x) = w_{q(x)}$, $q : \mathbb{R}^p \to \mathcal{L}$ reflects the structure of the tree f that maps an example to the corresponding leaf index, $\mathcal{L}$ is the set of the leaf indices, $w \in \mathbb{R}^T$ is leaf weight, and T is the number of leaves in the tree. Suppose that K regression trees in $\mathcal{F}$, $f_k(\cdot) \in \mathcal{F}$ with $k = 1, \cdots, K$, are used to predict the output:

$$\widehat{y}_{\cdot j} = \sum_{k=1}^{K} f_k(x_{\cdot j})$$

for an example with the input $x_{\cdot j}$.

To learn the set of functions used for classification, we minimize the regularized objective function

$$\mathcal{L}(y, \widehat{y}) = \sum_{j=1}^{n} L(y_{\cdot j}, \widehat{y}_{\cdot j}) + \sum_{k=1}^{K} \Omega(f_k), \tag{31}$$

where Ω is the regularization used to measure the model complexity, given by

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2 \tag{32}$$

with tuning parameters γ and λ. Here $L(\cdot)$ is the loss function which measures how well the model fits the training data. With the multiclass classification problem discussed in "Classification with predictor graphical structures accommodated" section, we specify $L(\cdot)$ as

$$\sum_{j=1}^{n} L(y_{\cdot j}, \widehat{y}_{\cdot j}) = -\sum_{i=1}^{I}\sum_{j=1}^{n} y_{ij} \log\left(p_{ij}\right)$$

with $p_{ij} = \frac{\exp(\widehat{y}_{ij})}{1+\sum_{l=1}^{I-1}\exp(\widehat{y}_{lj})}$ for $i = 1, \ldots, I-1$ and $p_{Ij} = 1 - \sum_{i=1}^{I-1} p_{ij}$.

While the formulation of the objective function in (31) is conceptually easy to balance the tradeoff between predictive accuracy and model complexity, minimizing the objective function (31) cannot be directly carried out using traditional optimization procedures. One approach is to invoke the gradient boosting tree algorithm iteratively to call for a second order approximation to the objective function. Specifically, at iteration t, we define

$$\widehat{y}_{\cdot j}^{(t)} = \sum_{k=1}^{t} f_k\left(x_{\cdot j}\right) = \widehat{y}_{\cdot j}^{(t-1)} + f_t\left(x_{\cdot j}\right)$$

with $\widehat{y}_{\cdot j}^{(0)} = 0$, and hence the objective function

$$\mathcal{L}^{(t)}(y, \widehat{y}) = \sum_{j=1}^{n} L\left(y_{\cdot j}, \widehat{y}_{\cdot j}^{(t)}\right) + \Omega(f_t). \tag{33}$$

Applying the second-order approximation to (33) gives

$$\mathcal{L}^{(t)}(y, \widehat{y}) \approx \sum_{j=1}^{n} \left\{ L\left(y_{\cdot j}, \widehat{y}_{\cdot j}^{(t-1)}\right) + g_j f_t\left(x_{\cdot j}\right) + \frac{1}{2} h_j f_t^2\left(x_{\cdot j}\right) \right\} + \Omega(f_t), \tag{34}$$

where g_j and h_j are the first and second order gradients of the loss function $L(y_{\cdot j}, \widehat{y}^{(t-1)})$ with respect to $\widehat{y}^{(t-1)}$, respectively.

Let $I_m = \left\{j : q(x_{\cdot j}) = m\right\}$ denote the instance set of leaf m. Then by (32), (34) becomes

$$\mathcal{L}^{(t)}(y, \widehat{y}) \approx \sum_{m=1}^{T} \left\{ \left(\sum_{j \in I_m} g_j\right) w_m + \frac{1}{2} \left(\sum_{j \in I_m} h_j + \lambda\right) w_m^2 \right\} + \gamma T. \tag{35}$$

For a given tree structure $q(\cdot)$, minimizing (35) gives the optimal weight w_m^* of leaf m and the optimal value of (35), respectively, given by

$$\widehat{w}_m = -\frac{\sum_{j \in I_m} g_j}{\sum_{j \in I_m} h_j + \lambda} \quad \text{and} \quad \widehat{\mathcal{L}}^{(t)} = -\frac{1}{2} \sum_{m=1}^{T} \frac{\left(\sum_{j \in I_m} g_j\right)^2}{\sum_{j \in I_m} h_j + \lambda} + \gamma T.$$

Numerical studies

In this section, we first conduct simulation studies to evaluate the performance of the proposed procedures in "Classification with predictor graphical structures accommodated" section, and then we apply the procedures to analyze a real dataset to illustrate their usage. The discussion is carried out in contrast to the classification methods reviewed in "Evaluation of the performance" section as well as the usual multiclass logistic regression model in "Logistic regression model for multiclass response" section. The R packages, `svm(e1071)`, `lda(MASS)`, `knn.cv(class)`, and `xgboost` are used to implement the SVM, LDA, KNN, and XGBOOST methods, respectively.

Simulation study

For class $i = 1, \cdots, I$, the predictors are generated from the multivariate normal distribution with mean zero and covariance matrix $\Sigma_i = \Omega_i^{-1}$, where Ω_i is a matrix associated with the network structure in class i with all diagonal elements 1 and off-diagonal elements 0 or 1; for $s \neq t$, entry (s, t) is 1 if the edge exists between X_s and X_t and 0 otherwise. The relationship between a multivariate normal distribution $N(0, \Sigma_i)$ and the *Gaussian graphical model* with edges determined by $\Omega_i = \Sigma_i^{-1}$ is discussed by Hastie et al. (2015, p.246 and p.263).

We specifically consider two scenarios of network structures where the dimension of predictors is $p = 12$. In the first scenario we specify Ω_i to reflect the network structures displayed in Fig. 1. For example, element $(1, 5)$ for Ω_1 is 1, but element $(1, 5)$ for Ω_i is 0 if $i = 2, 3, 4$. For a given class i and a subject j in this class, we calculate $\pi_j^i(x_{\cdot j})$ by (16) where we set $\gamma_0^i = \gamma_{st}^i = 1$. The outcome measurements are set to be $Y_j^i = 1$ if $\pi_j^i(x_{\cdot j}) > c$, and $Y_j^i = 0$ otherwise, where the threshold c is chosen such that the size in class i equals n_i.

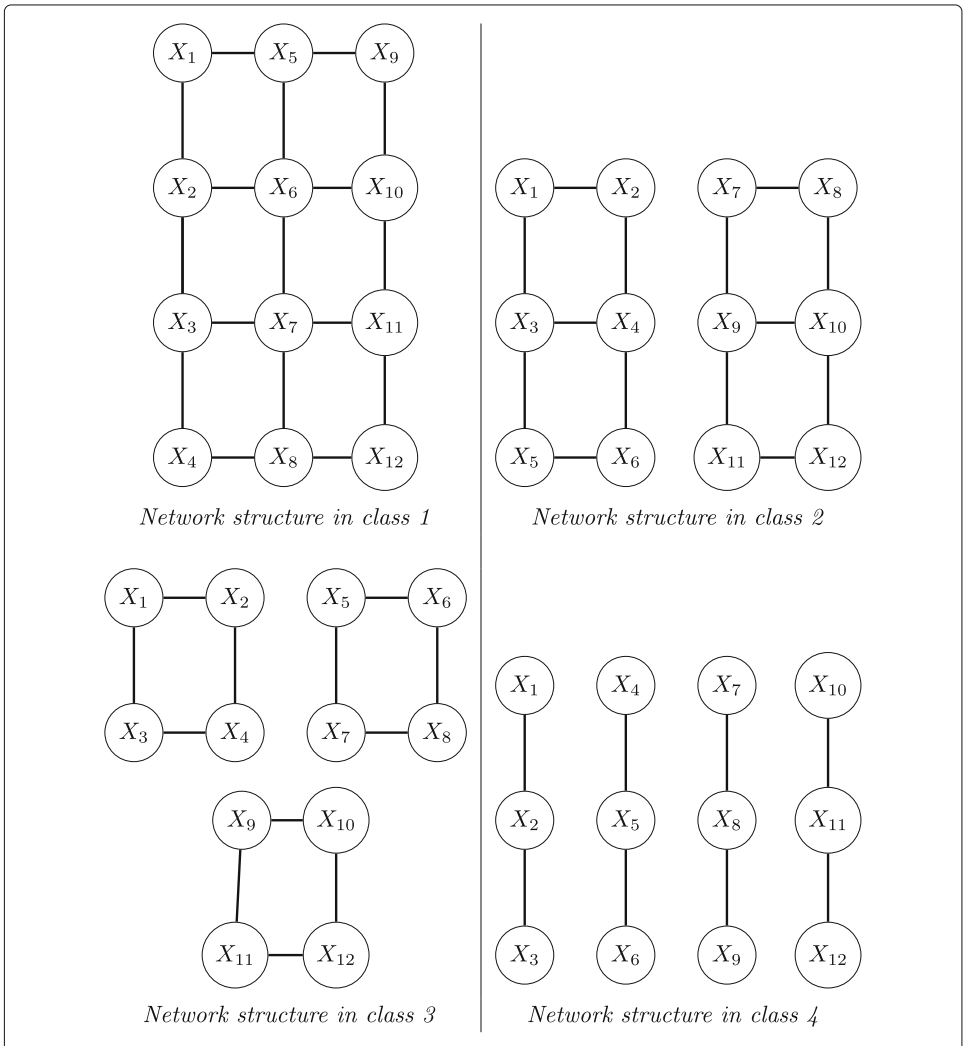

Fig. 1 Covariate network structures of four classes for the simulation study

In the second scenario, Ω_i is taken as the identity matrix for $i = 1, \cdots, I$, showing that the predictors have no network structures. For subject j, the predictor $X_{\cdot j}$ is generated from the multivariate normal distribution with mean zero and identity matrix. To generate $Y_{\cdot j}$ for subject j, we first calculate $\pi_{ij}(x_{\cdot j})$ for every $i = 1, \cdots, I$ by (2) and (3) where γ_{0i} and γ_i are both set as $\log(i) + 1$ for class i. Then we set $Y_{\cdot j} = i^*$ if $i^* = \underset{i}{\text{argmax}}\pi_{ij}(x_{\cdot j})$. Continue this process until the desired size n_i is achieved for $i = 1, \cdots, I$. We consider the case with $I = 4$ and $n_i = 50$ for $i = 1, \cdots, I$ and run 500 simulations. We use criteria (22) and (23) to report the performance of each method. The results are summarized in Table 2. It is seen that the proposed LR-ClassGraph method outperforms all the classification methods with larger values of *PRE*, *REC* and *F* from both micro and macro view points. The SVM performs the second best, and the performance of the LR-HomoGraph method is ranked the third, followed by that of the XGBOOST method.

To understand how the proposed methods perform with the binary classification, we repeat the preceding simulations by setting I to be 2 and taking the network structures of classes 1 and 2 when considering scenario 1. The results are in Table 3. When covariates are associated with a network structure, the proposed LR-ClassGraph method still performs the best, and the improvement of the LR-ClassGraph method over existing classifiers is a lot more noticeable for $I = 2$ than for $I = 4$. Interestingly, when covariates are uncorrelated, unlike the multiclass case with $I = 4$, the LR-HomoGraph method outperforms the LR-ClassGraph method; and in this case, the SVM is the best classifier.

Glass identification dataset

We analyze a dataset concerning glass identification. The study of classification of glass types was motivated by criminological investigation. At the scene of the crime, the glass left can be used as evidence if it is correctly identified. It is of interest to predict the glass type based on the information of the predictors.

The dataset contains 7 types of glass, including

- `building_windows_float_processed` (Glass-1),
- `building_windows_non_float_processed` (Glass-2),
- `vehicle_windows_float_processed` (Glass-3),

Table 2 Simulation study with and without network structures for covariates, respectively, indicated by Scenarios 1 and 2: $I = 4$

Scenario	Criteria	Agresti	SVM	LDA	KNN	XGBOOST	LR-HomoGraph	LR-ClassGraph
1	PRE_{micro}	0.635	0.830	0.640	0.678	0.690	0.841	0.890
	REC_{micro}	0.635	0.830	0.640	0.700	0.690	0.841	0.890
	F_{micro}	0.635	0.830	0.640	0.689	0.690	0.841	0.890
	PRE_{macro}	0.637	0.843	0.643	0.686	0.688	0.847	0.898
	REC_{macro}	0.635	0.830	0.640	0.704	0.690	0.842	0.891
	F_{macro}	0.636	0.836	0.641	0.695	0.689	0.844	0.894
2	PRE_{micro}	0.703	0.855	0.739	0.672	0.790	0.851	0.861
	REC_{micro}	0.717	0.855	0.734	0.672	0.790	0.851	0.866
	F_{micro}	0.710	0.855	0.736	0.672	0.790	0.851	0.863
	PRE_{macro}	0.706	0.805	0.740	0.704	0.792	0.859	0.860
	REC_{macro}	0.717	0.855	0.733	0.672	0.790	0.862	0.866
	F_{macro}	0.711	0.830	0.736	0.687	0.791	0.860	0.863

Table 3 Simulation study with and without network structures for covariates, respectively, indicated by Scenarios 1 and 2: $l = 2$

Scenario	Criteria	Agresti	SVM	LDA	KNN	XGBOOST	LR-HomoGraph	LR-ClassGraph
1	PRE_{micro}	0.625	0.835	0.625	0.685	0.615	0.825	0.965
	REC_{micro}	0.625	0.835	0.625	0.685	0.615	0.825	0.965
	F_{micro}	0.625	0.835	0.625	0.685	0.615	0.825	0.965
	PRE_{macro}	0.625	0.866	0.626	0.688	0.615	0.828	0.965
	REC_{macro}	0.626	0.835	0.625	0.685	0.615	0.825	0.965
	F_{macro}	0.625	0.850	0.626	0.686	0.615	0.825	0.965
2	PRE_{micro}	0.860	0.985	0.850	0.565	0.775	0.825	0.795
	REC_{micro}	0.860	0.985	0.850	0.565	0.775	0.825	0.795
	F_{micro}	0.860	0.985	0.850	0.565	0.775	0.825	0.795
	PRE_{macro}	0.861	0.981	0.605	0.850	0.775	0.820	0.770
	REC_{macro}	0.860	0.984	0.605	0.850	0.775	0.820	0.772
	F_{macro}	0.861	0.982	0.605	0.850	0.775	0.820	0.771

- `vehicle_windows_non_float_processed` (Glass-4),
- `containers` (Glass-5),
- `tableware` (Glass-6), and
- `headlamps` (Glass-7),

and the predictors include 9 different chemical materials, `refractive index (RI)`, `Sodium (NA)`, `Magnesium (MG)`, `Aluminum (AL)`, `Silicon (SI)`, `Potassium (K)`, `Calcium (CA)`, `Barium (BA)`, and `Iron (FE)`. The complete dataset is available at https://archive.ics.uci.edu/ml/datasets/glass+identification. The sample size in each class is, respectively, $n_1 = 70$, $n_2 = 76$, $n_3 = 17$, $n_4 = 0$, $n_5 = 13$, $n_6 = 9$, and $n_7 = 29$, yielding the total sample size $n = \sum_{i=1}^{7} n_i = 214$. To see the correlation among the predictors, we draw a scatter plot of those 9 predictors, displayed in Fig. 2. It is seen that some predictors, such as `RI` and `CA`, are highly correlated, and that many pairwise predictors are generally correlated.

We first present the network structures for different chemical materials in each class. The network structure for each class is determined by (9) and (11). The graphical results

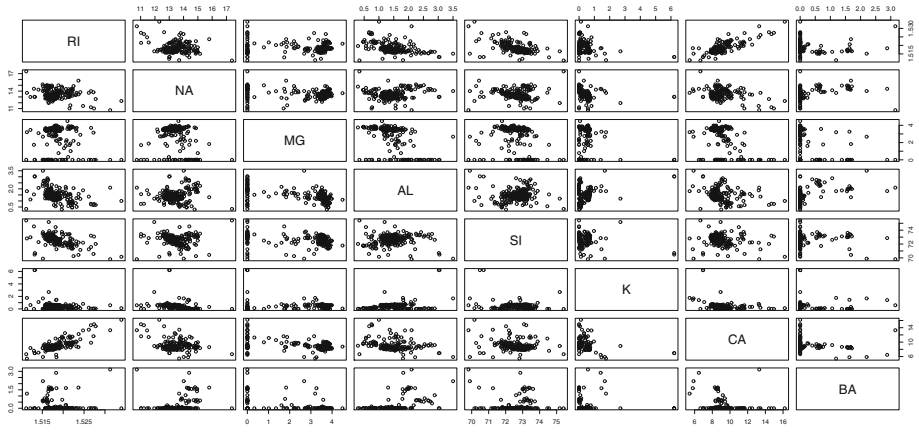

Fig. 2 Pairwise scatter plots of predictors in glass data

are reported in Fig. 4. It is seen that the network structure of the predictors is different from class to class. We notice that `RI` has no connection with other variables in every class and the predictor `FE` also has no connection with others except in class 6.

We next evaluate the performance of our proposed methods as opposed to the conventional approaches, SVM, LDA, KNN, and XGBOOST, which are respectively implemented by the R packages `svm(e1071)`, `lda(MASS)`, `knn.cv(class)`, and `xgboost`. To examine the performance of LR-HomoGraph proposed in "Logistic regression with homogeneous graphically structured predictors" section, we first construct the network structures, displayed in Fig. 3, of the predictors with the class information ignored, and we then apply the procedure described in "Logistic regression with homogeneous graphically structured predictors" section. To implement the LR-ClassGraph method in "Logistic regression with class-dependent graphically structured predictors" section, we apply model (16) with respect to six different network structures in Fig. 4, and then determine the predictive class using (18).

To measure the classification results in each class, we define the *misclassification rate in class i* to be

$$\mathrm{MIS}_i = \frac{1}{n_i}\sum_{j=1}^{n} \mathbb{I}\left(y_{\cdot j} = i, \widehat{y}_{\cdot j} \neq i\right) \quad \text{for } i = 1, \cdots, I.$$

The results obtained from SVM, LDA, KNN, XGBOOST, and the proposed methods are reported in Table 4. The misclassification rate of our proposed methods in each class are smaller than other methods, and the LR-ClassGraph yields the smallest misclassification

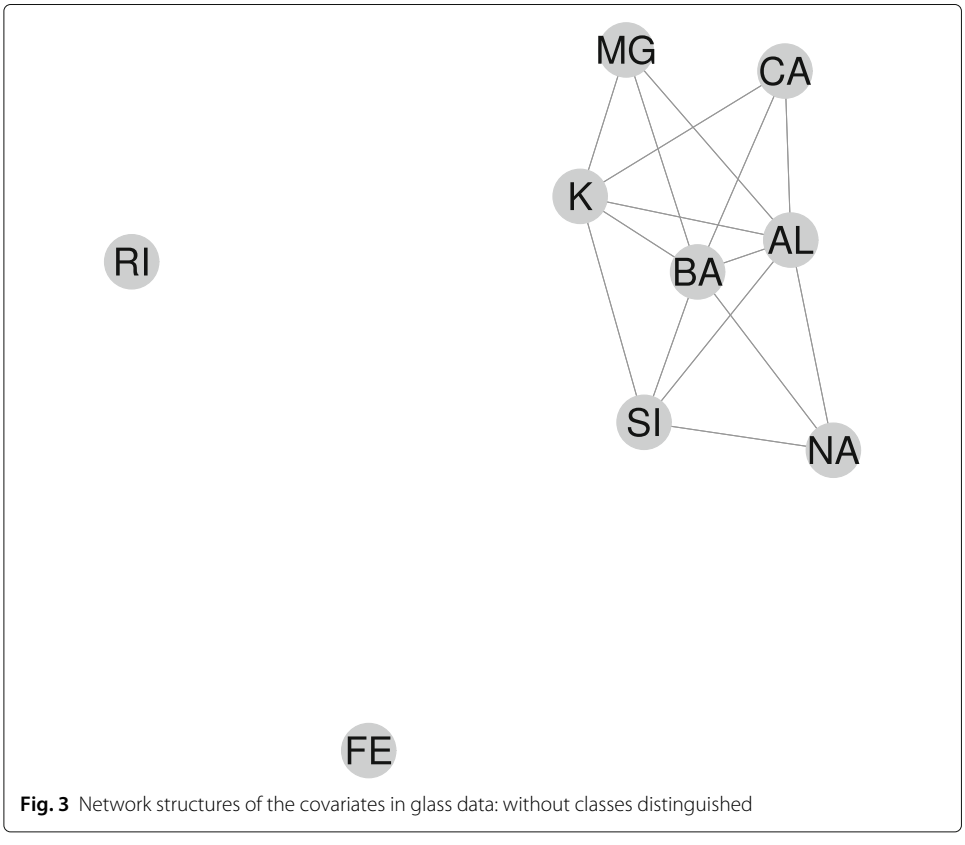

Fig. 3 Network structures of the covariates in glass data: without classes distinguished

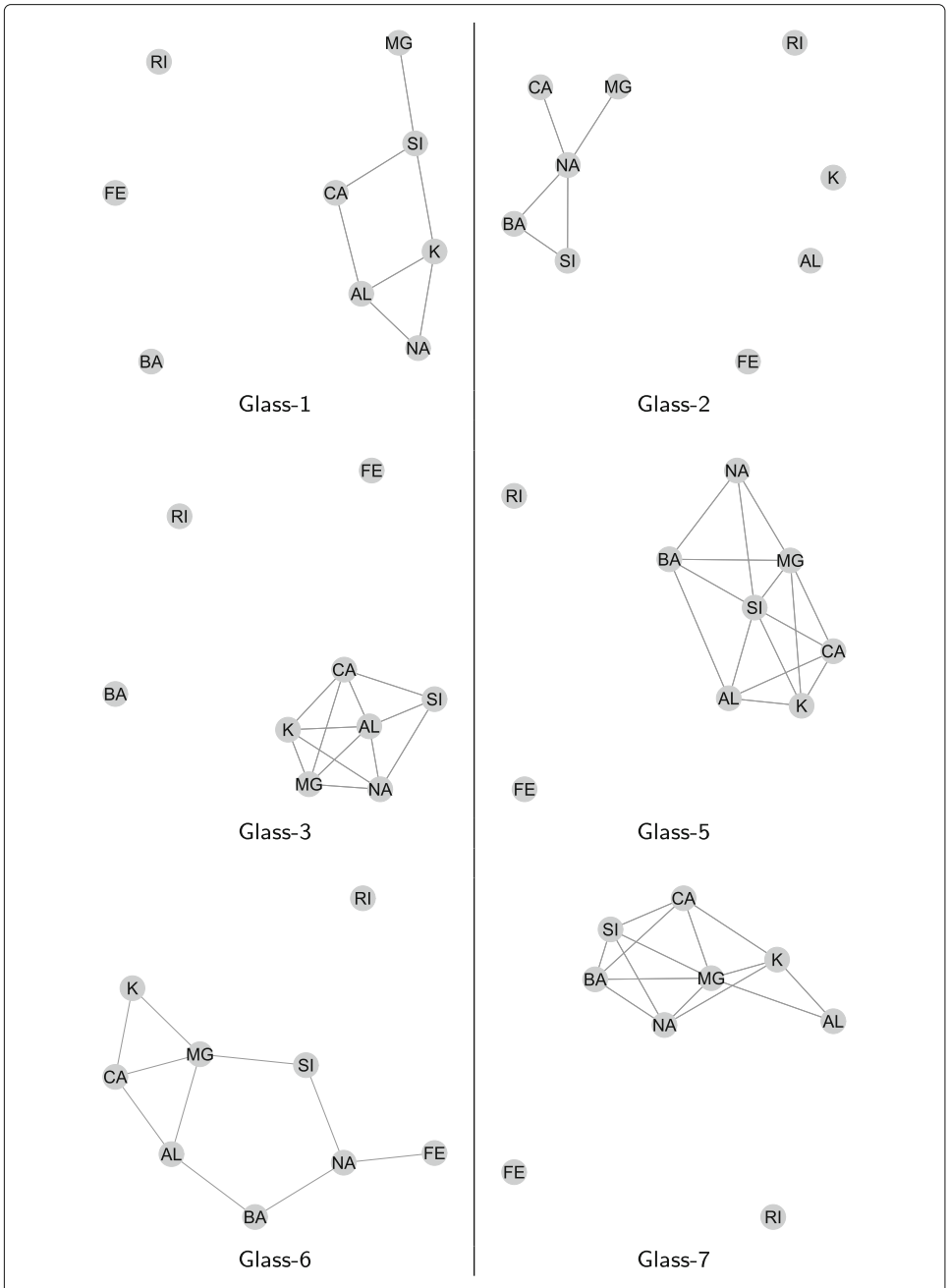

Fig. 4 Network structures of the covariates in glass data: within each class

rate for each class. Among the four compared methods, the SVM outperforms the other three methods.

Finally, we use criteria (22) and (23) to compare the overall performance of all the methods and summarize the results in Table 5. It is clear that both LR-HomoGraph and LR-ClassGraph produce higher values of the *F*, *PRE* and *REC* measures, regardless of micro and macro, implying that our proposed methods perform better than other multiclassification methods considered here. In addition, we further implement the two methods in "Classification with predictor graphical structures accommodated" section by respectively extending models (12) and (17) with the linear terms in each predictor included, and

Table 4 Classification results for glass data

		Glass-1	Glass-2	Glass-3	Glass-5	Glass-6	Glass-7
Agresti	Glass-1	52	19	10	0	0	0
	Glass-2	18	54	7	3	0	2
	Glass-3	0	0	0	0	0	0
	Glass-5	0	1	0	9	0	0
	Glass-6	0	0	0	0	9	0
	Glass-7	0	2	0	1	0	27
	MIS_i	0.257	0.289	1.000	0.307	0.000	0.069
SVM	Glass-1	59	15	10	0	0	1
	Glass-2	11	61	7	0	1	0
	Glass-3	0	0	0	0	0	0
	Glass-5	0	0	0	13	0	0
	Glass-6	0	0	0	0	8	0
	Glass-7	0	0	0	0	0	28
	MIS_i	0.157	0.197	1.000	0.000	0.111	0.034
LDA	Glass-1	46	16	3	0	1	0
	Glass-2	14	41	3	2	1	1
	Glass-3	10	12	11	0	0	1
	Glass-5	0	4	0	10	0	2
	Glass-6	0	3	0	0	7	1
	Glass-7	0	0	0	1	0	24
	MIS_i	0.343	0.461	0.353	0.231	0.222	0.172
KNN	Glass-1	51	17	14	0	0	1
	Glass-2	12	52	1	3	2	2
	Glass-3	7	2	2	0	0	0
	Glass-5	0	3	0	8	0	1
	Glass-6	0	2	0	0	4	2
	Glass-7	0	0	0	2	3	22
	MIS_i	0.271	0.316	0.882	0.385	0.556	0.241
XGBOOST	Glass-1	56	7	7	0	0	1
	Glass-2	8	61	5	6	2	0
	Glass-3	5	2	4	0	0	0
	Glass-5	0	4	0	6	1	2
	Glass-6	1	1	1	0	6	0
	Glass-7	0	1	0	1	0	26
	MIS_i	0.200	0.197	0.765	0.538	0.333	0.103
LR-HomoGraph	Glass-1	60	15	8	0	0	1
	Glass-2	10	60	6	0	0	0
	Glass-3	0	1	3	0	0	0
	Glass-5	0	0	0	13	0	0
	Glass-6	0	0	0	0	9	0
	Glass-7	0	0	0	0	0	28
	MIS_i	0.143	0.211	0.824	0.000	0.000	0.034
LR-ClassGraph	Glass-1	61	10	8	0	0	1
	Glass-2	9	66	6	0	0	0
	Glass-3	0	0	3	0	0	0
	Glass-5	0	0	0	13	0	0
	Glass-6	0	0	0	0	9	0
	Glass-7	0	0	0	0	0	28
	MIS_i	0.129	0.132	0.824	0.000	0.000	0.034

Table 5 Overall performance of classification methods applied to glass data

	Agresti	SVM	LDA	KNN	XGBOOST	LR-HomoGraph	LR-ClassGraph	LR-HomoGraph +main	LR-ClassGraph +main
PRE_{micro}	0.706	0.790	0.649	0.653	0.677	0.808	0.841	0.776	0.783
REC_{micro}	0.706	0.790	0.650	0.653	0.730	0.808	0.841	0.776	0.794
F_{micro}	0.706	0.790	0.649	0.653	0.703	0.808	0.841	0.776	0.788
PRE_{macro}	0.681	0.743	0.651	0.583	0.615	0.876	0.929	0.817	0.816
REC_{macro}	0.680	0.755	0.703	0.563	0.644	0.800	0.814	0.774	0.816
F_{macro}	0.680	0.749	0.676	0.573	0.629	0.836	0.868	0.795	0.816

we denote those methods as LR-HomoGraph+main and LR-ClassGraph+main, respectively, and report the results in the last two columns of Table 5. Such an extension of the models, however, does not help increase the values of these measures.

Discussion

In this paper, we propose to use logistic regression methods to make a prediction for data with network structures in predictors. In our methods, we first identify the network structures of the predictors for every class using graphical models, and then we capitalize on the identified network structures for the predictors to fit a logistic regression model to do classification and prediction. Simulation studies demonstrate that in the presence of network structures for covariates, our proposed methods produce more precise classification results than conventional methods, such as SVM, LDA, KNN, and XGBOOST. To allow interested readers to use the algorithms developed in "Classification with predictor graphical structures accommodated" section, the implementation procedures will be posted at CRAN.

Our development here focuses on examining pairwise dependence structures among predictors using the formulation (7). This is primarily driven by the consideration that such a dependence structure is intuitively interpretable and commonly exists in many problems. Extensions to facilitating triplewise or higher order dependence structures or even with the main effects (i.e., single variable effects), among predictors can be carried out by extending (7) to the form (9.5) of Hastie et al. (2015). Such extensions are, in principle, straightforward to implement technically, but the issue of overfitting may arise. In addition, underlying constraints on the model parameters may become a complex concern in numerical implementation. Discussions on this aspect were given by many authors, including Yang et al. (2015), Yi (2017), and Yi et al. (2017). Our discussion in this paper is directed to using the exponential family distribution to facilitate continuous predictor. It is easy to extend our methods to accommodate mixture graphical models which feature both continuous and discrete predictors.

In obtaining the estimator (9), we use the L_1-norm or the LASSO penalty, which is driven by its popularity as well as the availability of the implementation software packages (e.g., R packages *huge* and *XMRF*). However, the methods described in "Classification with predictor graphical structures accommodated" section are not just confined to the LASSO penalty. Our methods apply as well when other penalty functions are used. For instance, penalty functions, such as the elastic-net, SCAD, adaptive LASSO, L_2-norm penalties can be used to replace the LASSO penalty in deriving the estimator (9); the remaining procedures developed in "Classification with predictor graphical structures

accommodated" section still carry through. It will be interesting to conduct numerical studies for the use of different penalty functions to compare how results may differ with and without incorporating the network structure in the analysis, as noted by a referee. Though in this paper we are not able to exhaust numerical explorations for all possible penalty functions, the implementation framework presented in "Classification with predictor graphical structures accommodated" section allows the users to take any penalty functions that suit their own problems.

Finally, we comment that several aspects of the methods described in "Classification with predictor graphical structures accommodated" section warrants further research. As pointed out by a referee, our methods are developed for the problems with low dimensional data (i.e., $p < n$) and they are not applicable to sizable data with $p \geq n$. In the current digital world, it is not uncommon that we often have to handle data with thousands of predictor variables but the sample size is a lot smaller. In such circumstances, dimension reduction or feature screening techniques would be employed before proceeding with formal data analysis. It is interesting to generalize our methods to handle high-dimensional data with p being of a polynomial order of n or even ultra high-dimensional data with p being of an exponential order of n.

Our methods basically involve two steps in using measurements for the covariates and class labels. In the first step, we utilize *undirected graphs* to examine the covariate measurements alone, and the class information only comes into play in the second step when using logistic regression for classification. Alternatively, one may consider using *directed acyclic graphs* to feature conditional independencies among variables and develop probabilistic graphical models for classification. To evaluate the performance of the proposed methods, we focus on the comparisons with the competing classifiers reviewed in "Evaluation of the performance" section. While those algorithms cover a good range of available classifiers, they are not exhaustive, or even far from being comprehensive, in comparisons. Despite the frequentist nature of our methods, it is interesting to compare the proposed methods to the Bayesian network classifiers which have proven useful in applications (e.g., Geiger and Heckerman 1996; Pérez et al. 2006; Bielza and Larrañaga 2014). Furthermore, it is worthwhile to employ rigorous hypothesis testing procedures to evaluate whether the differences in the results obtained from different classifiers are statistically significant.

Acknowledgements
This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and partially supported by a Collaborative Research Team Project of the Canadian Statistical Sciences Institute (CANSSI).

Authors' contributions
The first two authors lead the project with equal contributions including writing the paper; the last two authors participate in the project with equal contributions.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Author details
[1] Department of Statistics and Actuarial Science, University of Waterloo, 200 University Ave W, N2L 3G1 Waterloo, Canada . [2] Department of Statistical and Actuarial Sciences, University of Western Ontario, 1151 Richmond St North, N6A 5B7 London, Canada .

Received: 25 October 2018 Accepted: 6 May 2019
Published online: 06 June 2019

References

Agresti, A.: An Introduction to Categorical Data Analysis. Wiley, New York (2007)
Agresti, A.: Categorical Data Analysis. Wiley, New York (2012)
Bagirov, A. M., Ferguson, B., Ivkovic, S., Saunders, G., Yearwood, J.: New algorithms for multi-class cancer diagnosis using tumor gene expression signatures. Bioinformatics. **19**, 1800–1807 (2003)
Baladanddayuthapani, V., Talluri, R., Ji, Y., Coombes, K. R., Lu, Y., Hennessy, B. T., Davies, M. A., Mallick, B. K.: Bayesian sparse graphical models for classification with application to protein expression data. Ann. Appl. Stat. **8**, 1443–1468 (2014)
Bicciato, S., Luchini, A., Bello, C. D.: Pca disjoint models for multiclass cancer analysis using gene expression data. Bioinformatics. **19**, 571–578 (2003)
Bielza, C., Li, G., Larrañaga, P.: Multi-dimensional classification with bayesian networks. Int. J. Approx. Reason. **52**, 705–727 (2011)
Bielza, C., Larrañaga, P.: Discrete bayesian network classifiers: A survey. ACM Comput. Surv. **47**, 1–43 (2014)
Cai, W., Guan, G., Pan, R., Zhu, X., Wang, H.: Network linear discriminant analysis. Comput. Stat. Data Anal. **117**, 32–44 (2018)
Cetiner, M., Akgul, Y. S.: Information Sciences and Systems 2014. In: In: T., C., E., G., R., L. (eds.) 2nd, pp. 53–76. Springer, New York, (2014)
Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of KDD '16 Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. ACM, San Francisco, (2016). http://doi.org/10.1145/2939672.2939785
Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, Cambridge (2000)
Fan, J., Li, R.: Variable selection via nonconcave penalized likelihood and its oracle properties. J. Am. Stat. Assoc. **96**, 1348–1360 (2001)
Friedman, J., Hastie, T., Tibshirani, R.: Sparse inverse covariance estimation with the graphical lasso. Biostatistics. **9**, 432–441 (2008)
Geiger, D., Heckerman, D.: Knowledge representation and inference in similarity networks and bayesian multinets. Artif. Intell. **82**, 45–74 (1996)
Guo, Y., Hastie, T., Tibshirani, R.: Regularized linear discriminant analysis and its application in microarrays. Biostatistics. **8**, 86–100 (2007)
Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York (2008)
Hastie, T., Tibshirani, R., Wainwright, M.: Statistical Learning with Sparsity: The Lasso and Generalizations. CRC press, New York (2015)
Hsu, C.-W., Lin, C.-J.: A comparison of methods for multiclass support vector machines. IEEE Trans. Neural Netw. **13**, 415–425 (2002)
Huttenhower, C., Flamholz, A. I., Landis, J. N., Sahi, S., Myers, C. L., Olszewski, K. L., Hibbs, M. A., Siemers, N. O., Troyanskaya, O. G., Coller, H. A.: Nearest neighbor networks: clustering expression data based on gene neighborhoods. BMC Bioinformatics. **8**, 1–13 (2007)
James, G., Witten, D., Hastie, T., Tibshirani, R.: An Introduction to Statistical Learning: with Applications in R. Springer, New York (2017)
Knerr, S., Personnaz, L., Dreyfus, G.: Single-layer learning revisited: A stepwise procedure for building and training neural network. In: In: F.F., S., J., H. (eds.) Neurocomputing: Algorithms, Architectures and Applications. 1st, pp. 41–50. Springer, Berlin, (1990)
Lee, Y., Lee, C.-K.: Classification of multiple cancer types by multicategory support vector machines using gene expression data. Bioinformatics. **19**, 1132–1139 (2003)
Lee, J., Hastie, T. J.: earning the structure of mixed graphical models. J. Comput. Graph. Stat. **24**, 230–253 (2015)
Liu, J. J., Cutler, G., Li, W., Pan, Z., Peng, S., Hoey, T., Chen, L., Ling, X. B.: Multiclass cancer classification and biomarker discovery using ga-based algorithms. Bioinformatics. **21**, 2691–2697 (2005)
Meinshausen, N., Bühlmann, P.: High-dimensional graphs and variable selection with the lasso. Ann. Stat. **34**, 1436–1462 (2006)
Miguel Hernández-Lobato, J., Hernández-Lobato, D., Suárez, A.: Network-based sparse bayesian classification. Pattern Recognit. **44**, 886–900 (2011)
Parambath, S. A. P., Usunier, N., Grandvalet, Y.: Optimizing pseudo-linear performance measures: Application to f-measure (2018). arXiv:1505.00199v4. Accessed 1 Jan 2018
Pérez, A., Larrañaga, P., Inza, I.: Supervised classification with conditional gaussian networks: Increasing the structure complexity from naive bayes. Int. J. Approx. Reason. **43**, 1–25 (2006)
Peterson, C. B., Stingo, F. C., Vannucci, M.: Joint bayesian variable and graph selection for regression models with network-structured predictors. Stat. Med. **35**, 1017–1031 (2015)
Ravikumar, P., Wainwright, M. J., Lafferty, J.: High-dimensional ising model selection using ℓ_1-regularized logistic regression. Ann. Stat. **38**, 1287–1319 (2010)
Safo, S. E., Ahn, J.: General sparse multi-class linear discriminant analysis. Comput. Stat. Data Anal. **99**, 81–90 (2016)
Sokolova, M., Japkowicz, N., Szpakowicz, S.: AI 2006: Advances in Artificial Intelligence. In: In: A., S., B., K. (eds.) 1st, pp. 53–76. Springer, Berlin, (2006)
Tibshirani, R.: Regression shrinkage and selection via the lasso. J. Royal Stat. Soc. Ser. B. **58**, 267–288 (1996)
Wang, H., Li, R., Tsai, C.: uning parameter selectors for the smoothly clipped absolute deviation method. Biometrika. **94**, 553–568 (2007)
Yang, E., Ravikumar, P., Allen, G. I., Liu, Z.: Graphical models via univariate exponential family distribution. J. Mach. Learn. Res. **16**, 3813–3847 (2015)

Yi, G. Y.: Composite likelihood/pseudolikelihood. Wiley StatsRef: Stat. Ref. Online (2017). https://doi.org/10.1002/9781118445112.stat07855
Yi, G. Y., He, W., Li, H.: A class of flexible models for analysis of complex structured correlated data with application to clustered longitudinal data. Stat. **6**, 448–461 (2017)
Zhu, S. X. Y., Pan, W.: Network-based support vector machine for classification of microarray samples. BMC Bioinformatics. **10**, 1–11 (2009)
Zi, X., Liu, Y., Gao, P.: Mutual information network-based support vector machine for identification of rheumatoid arthritis-related genes. Int. J. Clin. Experiment. Med. **9**, 11764–11771 (2016)
Zou, H.: The adaptive lasso and its oracle properties. J. Am. Stat. Assoc. **101**, 1418–1429 (2006)