Statistics and Its Interface Volume 13 (2020) 127–137

Bayesian kernel adaptive grouping learning for multi-dimensional datasets*

Xiaozhou Wang† and Fangli Dong†‡

With the development of information technology, a large number of datasets with complex structures, such as multi-dimensional datasets, need to be processed and analyzed. In this paper we propose a kernel-based statistical learning algorithm, Bayesian Kernel Adaptive Grouping Learning (BKAGL), to provide an innovative solution for the classification problem of multi-dimensional datasets. BKAGL can integrate information from different dimensions adaptively. Meanwhile, we adopt the Bayesian framework which can infer the approximate posterior distributions of parameters. The utilization of grouping features can help find which groups of features have more contributions to the response. Simulation results illustrate that BKAGL outperforms some classical classification methods and the corresponding ungrouped method. The analysis of the electrocardiogram (ECG) and electroencephalography (EEG) datasets shows that BKAGL has the highest classification accuracy and provides explanatory information.

AMS 2000 subject classifications: 62F15, 62H30.
Keywords and phrases: Classifier, Multi-dimensional dataset, Bayesian model, Adaptiveness, Kernel method.

1. INTRODUCTION

One of the main purposes of machine learning is to infer the relationship between the outcome Y and the variable X so as to improve human awareness [1, 7, 16]. However, with the explosive growth of the data, how to mine and integrate the information from the data is a challenge to the learning algorithms.

Support Vector Machine (SVM) [13] is a well-known machine learning method due to its excellent generalization properties. The decision function which is used for predicting the label y of one instance $\boldsymbol{x}$ can be written as

$$f(\boldsymbol{x}) = \boldsymbol{a}^{\mathrm{T}}\boldsymbol{k}(\boldsymbol{x}) + e, \tag{1}$$

where $\boldsymbol{a} = (a_1, ..., a_N)^{\mathrm{T}}$ is the sample weight, e is the bias, $\boldsymbol{k}(\boldsymbol{x}) = (k(\boldsymbol{x}_1, \boldsymbol{x}), ..., k(\boldsymbol{x}_N, \boldsymbol{x}))^{\mathrm{T}}$ and the kernel function $k(\boldsymbol{x}_i, \boldsymbol{x})$ satisfies Mercer's condition [10]. The parameters in Eq. (1) can be obtained through a quadratic optimization process [27]. Furthermore, the parameters can also be optimized under a Bayesian framework [26], which is called Relevance Vector Machine (RVM). The sample weight parameter $\boldsymbol{a}$ is viewed as a random vector. Both SVM and RVM are kernel-based methods which have a lot of advantages [14, 25]. Moreover, when we want to integrate data from different sources, the multiple kernel learning algorithm [12, 22, 24] is usually considered. For example, we use multiple omics data, including the high-throughput genomic, epigenomic, and transcriptomic data, to predict a person's phenotypic response [29]. From the viewpoint of data structure, multiple omics data correspond to multiple dimensions. For the aim of integrating these multi-dimensional data, multiple kernels can be used where each kernel is a function of each dimension. Then multiple kernels can integrate multi-dimensional data by a kernel-based decision function as follows:

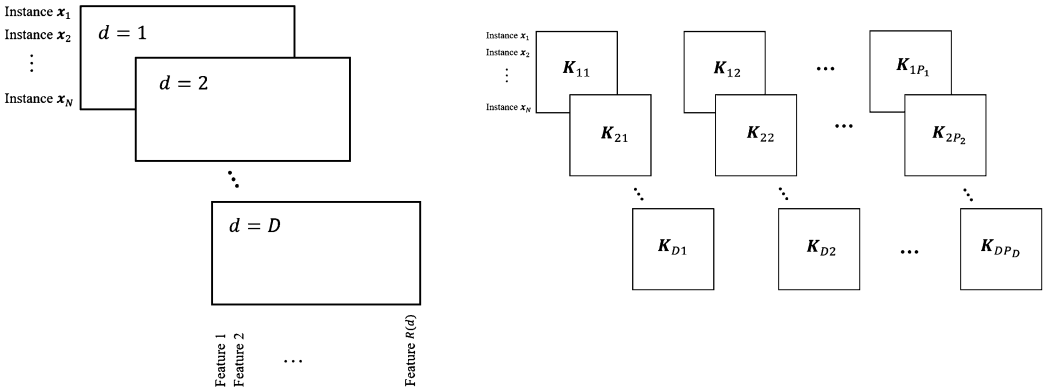
$$f(\boldsymbol{x}) = \boldsymbol{a}^{\mathrm{T}} \sum_{q=1}^{Q} \rho_q \boldsymbol{k}_q(\boldsymbol{x}) + e, \tag{2}$$

where Q is the number of kernels and also the number of dimensions, and $\boldsymbol{\rho} = (\rho_1, ..., \rho_Q)^{\mathrm{T}}$ is the kernel weight parameter. For the q-th kernel, $\boldsymbol{k}_q(\boldsymbol{x}) = (k_q(\boldsymbol{x}_1, \boldsymbol{x}), ..., k_q(\boldsymbol{x}_N, \boldsymbol{x}))^{\mathrm{T}}$. Through the estimation of kernel weight parameter, we can obtain which dimensions contribute more to the response. Moreover, for single omics data such as gene expression data, a pathway contains a group of genes. There are different pathways (groups) in gene expression data. If we want to decipher the correlation between the pathways and response, Eq. (2) can also be used through defining multiple kernels based on different pathways. However, a mass of parameters will need to be estimated. In this case, a novel model is necessary to integrate information from different groups in different dimensions. Naturally, the extension of Eq. (2) as below can be utilized.

$$f(\boldsymbol{x}) = \boldsymbol{a}^{\mathrm{T}} \sum_{d=1}^{D} c_d \sum_{m=1}^{P_d} b_{md} \boldsymbol{k}_{dm}(\boldsymbol{x}) + e, \tag{3}$$

where D is the number of dimensions and $\boldsymbol{c} = (c_1, ..., c_D)^{\mathrm{T}}$

*Supported by RGC Competitive Earmarked Research Grants, National Basic Research Program of China (973 Program, 2015CB856004) and National Natural Science Foundation of China (11531001).
†These authors contributed equally to this work.
‡Corresponding author.



(a) $\boldsymbol{x}$ (b) $\boldsymbol{K}$

Figure 1. Structures of the dataset $\boldsymbol{x}$ and kernel matrix set $\boldsymbol{K}$.

is the dimension weight parameter. P_d is the group number of features in the d-th dimension and $\boldsymbol{b}$ is the two-dimensional group weight parameter. For the m-th group in the d-th dimension, $\boldsymbol{k}_{dm}(\boldsymbol{x}) = (k_{dm}(\boldsymbol{x}_1, \boldsymbol{x}), ..., k_{dm}(\boldsymbol{x}_N, \boldsymbol{x}))^{\mathrm{T}}$ and $k_{dm}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is the kernel function of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ for the subjects i and j. Since the grouping result of each dimension may be different, the group number P_d and the parameter b_{md} are decided by the corresponding d-th dimension. As shown in Eq. (3), we propose a model which extracts information layer by layer. In particular, the proposed algorithm is based on a Bayesian framework [19, 21] and uses an approximate inference process [6]. BKAGL can simultaneously integrate information from different groups in different dimensions and then predict the outcome. At the same time, it has good interpretation ability, which enables us to find which dimensions and groups of features have higher contributions to the outcome. This is our concern.

The rest of the paper is organized as follows: Section 2 firstly introduces the notations used in the paper, and then explains the model. In Section 3, the process of parameter estimation and the prediction for a new point are provided. Section 4 examines the performance of our algorithm including accuracy and interpretability by the simulation and the experiment of the real ECG and EEG datasets. Conclusions and some extensions for further work are given in Section 5.

2. THE FRAMEWORK

2.1 Notations

Suppose that there are N i.i.d. instances $\{(\boldsymbol{x}_i, y_i) : i = 1, ..., N\}$. Each instance $\boldsymbol{x}_i$ has D dimensions and each dimension is a vector with $R(d)$ features. More precisely, x_{idr} represents the r-th feature of the d-th dimension of the i-th instance. The response y_i is the class label of the i-th instance $\boldsymbol{x}_i$. Assume that all the features of each dimension have been divided into different groups in a certain way. Denote P_d as the number of the groups of features in the d-th dimension. Define

- Kernel matrix set $\boldsymbol{K} = \{\boldsymbol{K}_{dm} : d = 1, ..., D, m = 1, ..., P_d\}$, where $\boldsymbol{K}_{dm}$ is a $N \times N$ kernel matrix of the m-th group of features in the d-th dimension. $(\boldsymbol{K}_{dm})_{\cdot i}$ is the i-th column of $\boldsymbol{K}_{dm}$ and $(\boldsymbol{K}_{dm})_{j\cdot}$ is the j-th row of $\boldsymbol{K}_{dm}$.
- $\boldsymbol{G} = \{\boldsymbol{G}_d : d = 1, ..., D\}$, where $\boldsymbol{G}_d$ is a $P_d \times N$ matrix. $\boldsymbol{G}_{d\cdot i}$ is the i-th column of $\boldsymbol{G}_d$ and $\boldsymbol{G}_{dm\cdot}$ is the m-th row of $\boldsymbol{G}_d$.
- $\boldsymbol{L}$ is a $D \times N$ matrix. $\boldsymbol{L}_{\cdot i}$ is the i-th column of $\boldsymbol{L}$ and $\boldsymbol{L}_{d\cdot}$ is the d-th row of $\boldsymbol{L}$.
- $\boldsymbol{f}$ is a vector with N elements. In fact, $\boldsymbol{f}$ is a latent variable connecting the second layer $\boldsymbol{L}$ of intermediate output and the class label vector $\boldsymbol{y}$ of N instances. More details about $\boldsymbol{f}$ are in Section 2.2.

Figure 1 shows the structures of dataset $\boldsymbol{x}$ and kernel matrix set $\boldsymbol{K}$ in order to help understand the data structure.

2.2 Model

The main aim of BKAGL is to add the interpretability of the model through grouping features and make the number of estimated parameters under control at the same time compared with Eq. (2). The proposed BKAGL algorithm extracts information from layer to layer. After getting the kernel matrix set $\boldsymbol{K}$ through the input training set, we calculate the first layer $\boldsymbol{G}$ of intermediate output by combining different kernels for different groups in each dimension as

$$\boldsymbol{G}_d = \begin{pmatrix} \boldsymbol{a}^{\mathrm{T}}\boldsymbol{K}_{d1} \\ \vdots \\ \boldsymbol{a}^{\mathrm{T}}\boldsymbol{K}_{dP_d} \end{pmatrix}, \quad d = 1, ..., D. \tag{4}$$

After obtaining the first layer $\boldsymbol{G}$ of intermediate output, the second layer $\boldsymbol{L}$ of intermediate output can be calculated.

$$\boldsymbol{L} = \begin{pmatrix} \boldsymbol{b}_{\cdot 1}^{\mathrm{T}}\boldsymbol{G}_1 \\ \vdots \\ \boldsymbol{b}_{\cdot D}^{\mathrm{T}}\boldsymbol{G}_D \end{pmatrix}, \tag{5}$$

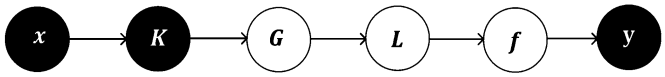


Figure 2. Classification of multi-dimensional datasets with BKAGL.

where $\boldsymbol{b}_{\cdot d} = (b_{1d}, ..., b_{P_d d})^{\mathrm{T}}$ for $d = 1, 2, ..., D$. Then through combining Eqs. (4) and (5), f_i, $i = 1, ..., N$, takes the following form:

$$\begin{aligned} f_i &= \boldsymbol{c}^{\mathrm{T}}\boldsymbol{L}_{\cdot i} + e = \sum_{d=1}^{D} c_d L_{di} + e \\ &= \sum_{d=1}^{D} c_d \boldsymbol{b}_{\cdot d}^{\mathrm{T}} \boldsymbol{G}_{d\cdot i} + e = \sum_{d=1}^{D} c_d \sum_{m=1}^{P_d} b_{md} G_{dmi} + e \\ &= \sum_{d=1}^{D} c_d \sum_{m=1}^{P_d} b_{md} \boldsymbol{a}^{\mathrm{T}} (\boldsymbol{K}_{dm})_{\cdot i} + e \\ &= \sum_{d=1}^{D} c_d \sum_{m=1}^{P_d} b_{md} \sum_{j=1}^{N} a_j (\boldsymbol{K}_{dm})_{ji} + e \\ &= \boldsymbol{a}^{\mathrm{T}} \sum_{d=1}^{D} \sum_{m=1}^{P_d} c_d b_{md} (\boldsymbol{K}_{dm})_{\cdot i} + e \triangleq \boldsymbol{a}^{\mathrm{T}} \boldsymbol{\mathcal{K}}_i + e. \end{aligned}$$

This formula indicates that the similarity measure $\boldsymbol{\mathcal{K}}_i$ is a linear combination of kernels defined in different groups in different dimensions. Through learning the parameters $\boldsymbol{b}$ and $\boldsymbol{c}$, we can find the more informative groups and dimensions for the response. The complete learning process is summarized in Figure 2.

For computational convenience, conjugate prior distributions [11] are applied in BKAGL. Denote all hyper-parameters $\boldsymbol{\Xi} = \{\alpha_\lambda, \beta_\lambda, \alpha_\eta, \beta_\eta, \alpha_\gamma, \beta_\gamma, \alpha_\omega, \beta_\omega\}$ and the prior set $\boldsymbol{\Theta} = \{\boldsymbol{\lambda}, \boldsymbol{\eta}, \boldsymbol{\gamma}, \omega\}$. Denote weight and bias parameters by the set $\boldsymbol{\Lambda} = \{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, e\}$, where vector $\boldsymbol{a}$ is the sample weight parameter, two-dimensional parameter $\boldsymbol{b}$ is the group weight parameter, vector $\boldsymbol{c}$ is the dimension weight parameter and e is the bias. Assume that the distributions of the related random variables in our proposed model (Eq. (3)) satisfy the following assumption (A1).

Assumption (A1): Note that $i = 1, ..., N$, $d = 1, ..., D$ and $m = 1, ..., P_d$. For parameters in $\boldsymbol{\Theta}$:

$$\begin{aligned} \lambda_i &\sim Gamma(\lambda_i; \alpha_\lambda, \beta_\lambda), \\ \eta_{md} &\sim Gamma(\eta_{md}; \alpha_\eta, \beta_\eta), \\ \gamma_d &\sim Gamma(\gamma_d; \alpha_\gamma, \beta_\gamma), \\ \omega &\sim Gamma(\omega; \alpha_\omega, \beta_\omega). \end{aligned}$$

For parameters in $\boldsymbol{\Lambda}$:

$$\begin{aligned} a_i|\lambda_i &\sim \mathcal{N}(a_i; 0, \lambda_i^{-1}), \\ b_{md}|\eta_{md} &\sim \mathcal{N}(b_{md}; 0, \eta_{md}^{-1}), \\ c_d|\gamma_d &\sim \mathcal{N}(c_d; 0, \gamma_d^{-1}), \\ e|\omega &\sim \mathcal{N}(e; 0, \omega^{-1}). \end{aligned}$$

For intermediate output variables:

$$\begin{aligned} G_{dmi}|\boldsymbol{a}, (\boldsymbol{K}_{dm})_{\cdot i} &\sim \mathcal{N}(G_{dmi}; \boldsymbol{a}^{\mathrm{T}}(\boldsymbol{K}_{dm})_{\cdot i}, 1), \\ L_{di}|\boldsymbol{b}_{\cdot d}, \boldsymbol{G}_{d\cdot i} &\sim \mathcal{N}(L_{di}; \boldsymbol{b}_{\cdot d}^{\mathrm{T}} \boldsymbol{G}_{d\cdot i}, 1). \end{aligned}$$

For the latent variable and the class label:

$$\begin{aligned} f_i|\boldsymbol{c}, e, \boldsymbol{L}_{\cdot i} &\sim \mathcal{N}(f_i; \boldsymbol{c}^{\mathrm{T}}\boldsymbol{L}_{\cdot i} + e, 1), \\ y_i|f_i &\sim \delta(f_i y_i > \tau), \end{aligned}$$

where τ is a given margin parameter relevant to the low density area of the distribution [15], $Gamma(\cdot; \alpha, \beta)$ denotes the gamma distribution with the mean $\alpha\beta$ and the variance $\alpha\beta^2$, $\mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ represents the normal distribution, and $\delta(\cdot)$ represents the Kronecker delta function that equals 1 if the condition is satisfied and 0 otherwise. In the process of training the model, we can tune the hyper-parameters $(\alpha_\lambda, \beta_\lambda)$, $(\alpha_\eta, \beta_\eta)$, and $(\alpha_\gamma, \beta_\gamma)$ to control the sparsity of the sample, group and dimension weights, respectively.

3. INFERENCE

3.1 Inference using variational approximation

In this section, we apply variational approximation [6] to the estimation of parameters. There are some advantages. For instance, there are closed-form iterative formulas and therefore the computational efficiency of variational approximation is superior to that of Markov Chain Monte Carlo (MCMC). Denote $\boldsymbol{\Phi} = \boldsymbol{\Lambda} \cup \{\boldsymbol{G}, \boldsymbol{L}, \boldsymbol{f}\}$. The exact posterior distribution is $p(\boldsymbol{\Theta}, \boldsymbol{\Phi}|\boldsymbol{y}, \boldsymbol{K})$ and has the factorable ensemble approximation $q(\boldsymbol{\Theta}, \boldsymbol{\Phi})$ [5] as follows:

$$\begin{aligned} p(\boldsymbol{\Theta}, \boldsymbol{\Phi}|\boldsymbol{y}, \boldsymbol{K}) \approx q(\boldsymbol{\Theta}, \boldsymbol{\Phi}) =& q(\boldsymbol{\lambda}) q(\boldsymbol{a}) q(\boldsymbol{G}) q(\boldsymbol{\eta}) q(\boldsymbol{b}) q(\boldsymbol{L}) q(\boldsymbol{\gamma}) \\ & \cdot q(\omega) q(e, \boldsymbol{c}) q(\boldsymbol{f}). \end{aligned}$$

This factored form of $q(\boldsymbol{\Theta}, \boldsymbol{\Phi})$ corresponds to mean field theory [20]. Then, the log marginal likelihood function can be calculated as

$$\begin{aligned} \log p(\boldsymbol{y}|\boldsymbol{K}) =& \mathrm{E}_{q(\boldsymbol{\Theta}, \boldsymbol{\Phi})}[\log p(\boldsymbol{y}|\boldsymbol{K})] \\ =& \int q(\boldsymbol{\Theta}, \boldsymbol{\Phi}) \Big[\log \frac{p(\boldsymbol{y}, \boldsymbol{\Theta}, \boldsymbol{\Phi}|\boldsymbol{K})}{q(\boldsymbol{\Theta}, \boldsymbol{\Phi})} \\ & - \log \frac{p(\boldsymbol{\Theta}, \boldsymbol{\Phi}|\boldsymbol{y}, \boldsymbol{K})}{q(\boldsymbol{\Theta}, \boldsymbol{\Phi})} \Big] d\boldsymbol{\Theta} d\boldsymbol{\Phi} \\ =& \mathcal{L}(\boldsymbol{\Theta}, \boldsymbol{\Phi}) + KL(q\|p), \end{aligned}$$

where

$$\mathcal{L}(\boldsymbol{\Theta},\boldsymbol{\Phi}) \triangleq \int q(\boldsymbol{\Theta},\boldsymbol{\Phi}) \log \frac{p(\boldsymbol{y},\boldsymbol{\Theta},\boldsymbol{\Phi}|\boldsymbol{K})}{q(\boldsymbol{\Theta},\boldsymbol{\Phi})} d\boldsymbol{\Theta} d\boldsymbol{\Phi}$$

and

$$KL(q\|p) \triangleq -\int q(\boldsymbol{\Theta},\boldsymbol{\Phi}) \log \frac{p(\boldsymbol{\Theta},\boldsymbol{\Phi}|\boldsymbol{y},\boldsymbol{K})}{q(\boldsymbol{\Theta},\boldsymbol{\Phi})} d\boldsymbol{\Theta} d\boldsymbol{\Phi}.$$

Note that the Kullback-Leibler divergence $KL(q\|p) \geq 0$, therefore the log marginal likelihood function satisfies

$$\begin{aligned}\log p(\boldsymbol{y}|\boldsymbol{K}) \geq \mathcal{L}(\boldsymbol{\Theta},\boldsymbol{\Phi}) = \mathrm{E}_{q(\boldsymbol{\Theta},\boldsymbol{\Phi})}[\log p(\boldsymbol{y},\boldsymbol{\Theta},\boldsymbol{\Phi}|\boldsymbol{K})] \\ - \mathrm{E}_{q(\boldsymbol{\Theta},\boldsymbol{\Phi})}[\log q(\boldsymbol{\Theta},\boldsymbol{\Phi})].\end{aligned}$$

Afterwards optimize the lower bound $\mathcal{L}(\boldsymbol{\Theta},\boldsymbol{\Phi})$ of the log marginal likelihood $\log p(\boldsymbol{y}|\boldsymbol{K})$. That means we need to solve an optimization problem as

$$\arg\max_{\boldsymbol{\Theta},\boldsymbol{\Phi}} \mathcal{L}(\boldsymbol{\Theta},\boldsymbol{\Phi}).$$

Then based on the calculus of variations [2], we know that each factor of $q(\boldsymbol{\Theta},\boldsymbol{\Phi})$ can be calculated from the following formula:

$$q(\cdot) \propto \exp(\mathrm{E}_{q(\boldsymbol{\Theta},\boldsymbol{\Phi}\setminus\cdot)}[\log p(\boldsymbol{y},\boldsymbol{\Theta},\boldsymbol{\Phi}|\boldsymbol{K})]). \tag{6}$$

Under the conjugate assumptions, we can compute the approximate posterior distribution of each parameter. First update the parameters in $\boldsymbol{\Theta}$ and $\boldsymbol{\Lambda}$, then the intermediate output variables $\{\boldsymbol{G},\boldsymbol{L}\}$ and finally the class label vector $\boldsymbol{y}$. The results are shown in Proposition 1.

Proposition 1. *Under the assumption (A1) and formula 6, the calculation results of the approximate posterior distributions are given below.*

$$\begin{aligned}
\boldsymbol{\lambda} &\sim \prod_{i=1}^{N} Gamma\left(\lambda_i;\alpha(\boldsymbol{\lambda}),\beta(\lambda_i)\right),\\
\boldsymbol{\eta} &\sim \prod_{d=1}^{D}\prod_{m=1}^{P_d} Gamma\left(\eta_{md};\alpha(\boldsymbol{\eta}),\beta(\eta_{md})\right),\\
\boldsymbol{\gamma} &\sim \prod_{d=1}^{D} Gamma\left(\gamma_d;\alpha(\boldsymbol{\gamma}),\beta(\gamma_d)\right),\\
\omega &\sim Gamma\left(\omega;\alpha(\omega),\beta(\omega)\right),\\
\boldsymbol{a} &\sim \mathcal{N}\left(\boldsymbol{a};\boldsymbol{\mu}(\boldsymbol{a}),\boldsymbol{\Sigma}(\boldsymbol{a})\right),\\
\boldsymbol{b} &\sim \prod_{d=1}^{D} \mathcal{N}\left(\boldsymbol{b}_{\cdot d};\boldsymbol{\mu}(\boldsymbol{b}_{\cdot d}),\boldsymbol{\Sigma}(\boldsymbol{b}_{\cdot d})\right),\\
\begin{pmatrix} e \\ \boldsymbol{c}\end{pmatrix} &\sim \mathcal{N}\left(\begin{pmatrix} e \\ \boldsymbol{c}\end{pmatrix};\boldsymbol{\mu}(e,\boldsymbol{c}),\boldsymbol{\Sigma}(e,\boldsymbol{c})\right),\\
\boldsymbol{G} &\sim \prod_{d=1}^{D}\prod_{i=1}^{N} \mathcal{N}\left(\boldsymbol{G}_{d\cdot i};\boldsymbol{\mu}(\boldsymbol{G}_{d\cdot i}),\boldsymbol{\Sigma}(\boldsymbol{G}_{d\cdot i})\right),\\
\boldsymbol{L} &\sim \prod_{i=1}^{N} \mathcal{N}\left(\boldsymbol{L}_{\cdot i};\boldsymbol{\mu}(\boldsymbol{L}_{\cdot i}),\boldsymbol{\Sigma}(\boldsymbol{L}_{\cdot i})\right),\\
\boldsymbol{f} &\sim \prod_{i=1}^{N} \mathcal{TN}\left(f_i;\widetilde{\boldsymbol{c}^{\mathrm{T}}}\widetilde{\boldsymbol{L}_{\cdot i}}+\tilde{e},1,f_i y_i>\tau\right),
\end{aligned}$$

where

$$\begin{aligned}
&\alpha(\boldsymbol{\lambda}) = \alpha_\lambda + \frac{1}{2}, \qquad \beta(\lambda_i) = \left(\beta_\lambda^{-1} + \frac{\widetilde{a_i^2}}{2}\right)^{-1},\\
&\alpha(\boldsymbol{\eta}) = \alpha_\eta + \frac{1}{2}, \qquad \beta(\eta_{md}) = \left(\beta_\eta^{-1} + \frac{\widetilde{b_{md}^2}}{2}\right)^{-1},\\
&\alpha(\boldsymbol{\gamma}) = \alpha_\gamma + \frac{1}{2}, \qquad \beta(\gamma_d) = \left(\beta_\gamma^{-1} + \frac{\widetilde{c_d^2}}{2}\right)^{-1},\\
&\alpha(\omega) = \alpha_\omega + \frac{1}{2}, \qquad \beta(\omega) = \left(\beta_\omega^{-1} + \frac{\widetilde{e^2}}{2}\right)^{-1},\\
&\boldsymbol{\mu}(\boldsymbol{a}) = \boldsymbol{\Sigma}(\boldsymbol{a})\sum_{d=1}^{D}\sum_{m=1}^{P_d}(\boldsymbol{K}_{dm})_{\cdot\cdot}\widetilde{\boldsymbol{G}_{dm\cdot}^{\mathrm{T}}},\\
&\boldsymbol{\Sigma}(\boldsymbol{a}) = \left(\mathrm{diag}(\tilde{\boldsymbol{\lambda}}) + \sum_{d=1}^{D}\sum_{m=1}^{P_d}(\boldsymbol{K}_{dm})_{\cdot\cdot}(\boldsymbol{K}_{dm})_{\cdot\cdot}^{\mathrm{T}}\right)^{-1},\\
&\boldsymbol{\mu}(\boldsymbol{b}_{\cdot d}) = \boldsymbol{\Sigma}(\boldsymbol{b}_{\cdot d})\widetilde{\boldsymbol{G}_{d\cdot\cdot}}\widetilde{\boldsymbol{L}_{d\cdot}^{\mathrm{T}}},\\
&\boldsymbol{\Sigma}(\boldsymbol{b}_{\cdot d}) = \left(\mathrm{diag}\left(\widetilde{\eta_{\cdot d}}\right) + \widetilde{\boldsymbol{G}_{d\cdot\cdot}\boldsymbol{G}_{d\cdot\cdot}^{\mathrm{T}}}\right)^{-1},\\
&\boldsymbol{\mu}(e,\boldsymbol{c}) = \boldsymbol{\Sigma}(e,\boldsymbol{c})\begin{pmatrix}\mathbb{1}^{\mathrm{T}}\tilde{\boldsymbol{f}} \\ \tilde{\boldsymbol{L}}\tilde{\boldsymbol{f}}\end{pmatrix},\\
&\boldsymbol{\Sigma}(e,\boldsymbol{c}) = \begin{pmatrix}\tilde{\omega}+N & \mathbb{1}^{\mathrm{T}}\widetilde{\boldsymbol{L}^{\mathrm{T}}} \\ \tilde{\boldsymbol{L}}\mathbb{1} & \mathrm{diag}(\tilde{\gamma}) + \widetilde{\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}}\end{pmatrix}^{-1},\\
&\boldsymbol{\mu}(\boldsymbol{G}_{d\cdot i}) = \boldsymbol{\Sigma}(\boldsymbol{G}_{d\cdot i})\left(\begin{pmatrix}(\boldsymbol{K}_{d1})_{i\cdot} \\ \vdots \\ (\boldsymbol{K}_{dP_d})_{i\cdot}\end{pmatrix}\tilde{\boldsymbol{a}} + \widetilde{\boldsymbol{b}_{\cdot d}}\widetilde{L_{di}}\right),\\
&\boldsymbol{\Sigma}(\boldsymbol{G}_{d\cdot i}) = \left(\boldsymbol{I} + \widetilde{\boldsymbol{b}_{\cdot d}\boldsymbol{b}_{\cdot d}^{\mathrm{T}}}\right)^{-1},\\
&\boldsymbol{\mu}(\boldsymbol{L}_{\cdot i}) = \boldsymbol{\Sigma}(\boldsymbol{L}_{\cdot i})(\tilde{\mathcal{G}} + \tilde{\boldsymbol{c}}\tilde{f}_i - \widetilde{\boldsymbol{c}e}),\\
&\boldsymbol{\Sigma}(\boldsymbol{L}_{\cdot i}) = \left(\boldsymbol{I} + \widetilde{\boldsymbol{c}\boldsymbol{c}^{\mathrm{T}}}\right)^{-1},\\
&\widetilde{\mathcal{G}} \triangleq \begin{pmatrix}\widetilde{\boldsymbol{b}_{\cdot 1}^{\mathrm{T}}}\widetilde{\boldsymbol{G}_{1\cdot i}} \\ \vdots \\ \widetilde{\boldsymbol{b}_{\cdot D}^{\mathrm{T}}}\widetilde{\boldsymbol{G}_{D\cdot i}}\end{pmatrix},
\end{aligned}$$

where $\widetilde{(\cdot)}$ *represents* $\mathrm{E}(\cdot)$*, the expectation operator.* $\mathbb{1}$ *is the all-one vector* $(1,1,...,1)^{\mathrm{T}}$*,* $\boldsymbol{I}$ *represents identity matrix and* $\mathcal{TN}$ *is the truncated normal distribution.*

Proof. The likelihood function and the approximate posterior distribution are

$$\begin{aligned} p(\boldsymbol{y},\boldsymbol{\Theta},\boldsymbol{\Phi}|\boldsymbol{K}) =& p(\boldsymbol{\lambda})p(\boldsymbol{a}|\boldsymbol{\lambda})p(\boldsymbol{G}|\boldsymbol{a},\boldsymbol{K})p(\boldsymbol{\eta})p(\boldsymbol{b}|\boldsymbol{\eta})p(\boldsymbol{L}|\boldsymbol{b},\boldsymbol{G}) \\ & \cdot p(\boldsymbol{\gamma})p(\boldsymbol{c}|\boldsymbol{\gamma})p(\omega)p(e|\omega)p(\boldsymbol{f}|\boldsymbol{c},e,\boldsymbol{L})p(\boldsymbol{y}|\boldsymbol{f}), \\ q(\boldsymbol{\Theta},\boldsymbol{\Phi}) =& q(\boldsymbol{\lambda})q(\boldsymbol{a})q(\boldsymbol{G})q(\boldsymbol{\eta})q(\boldsymbol{b})q(\boldsymbol{L})q(\boldsymbol{\gamma})q(\omega)q(e,\boldsymbol{c})q(\boldsymbol{f}). \end{aligned}$$

and each factor $q(\cdot)$ of the $q(\boldsymbol{\Theta},\boldsymbol{\Phi})$ can be calculated by formula 6. So we can first compute the approximate posterior distribution of the prior $\boldsymbol{\lambda}$ as below.

$$\begin{aligned} q(\boldsymbol{\lambda}) &\propto \exp\{\mathrm{E}_{q(\boldsymbol{\Theta},\boldsymbol{\Phi}\setminus\boldsymbol{\lambda})}\left[\log p(\boldsymbol{y},\boldsymbol{\Theta},\boldsymbol{\Phi}|\boldsymbol{K})\right]\} \\ &\propto \exp\{\log p(\boldsymbol{\lambda}) + \mathrm{E}_{q(\boldsymbol{a})}[\log p(\boldsymbol{a}|\boldsymbol{\lambda})]\} \\ &\propto \prod_{i=1}^{N}\left\{\lambda_i^{(\alpha_\lambda+\frac{1}{2})-1} e^{-\left(\beta_\lambda^{-1}+\frac{\mathrm{E}(a_i^2)}{2}\right)\lambda_i}\right\}. \end{aligned}$$

So far we obtain the approximate posterior distribution

$$q(\boldsymbol{\lambda}) = \prod_{i=1}^{N} Gamma\left(\lambda_i;\alpha_\lambda+\frac{1}{2},\left(\frac{1}{\beta_\lambda}+\frac{\mathrm{E}a_i^2}{2}\right)^{-1}\right).$$

Similarly, we can obtain $q(\boldsymbol{\eta})$, $q(\boldsymbol{\gamma})$ and $q(\omega)$.

Second, the approximate posterior distribution of the sample weight parameter can be calculated as

$$\begin{aligned} q(\boldsymbol{a}) \propto& \exp\{\mathrm{E}_{q(\boldsymbol{\Theta},\boldsymbol{\Phi}\setminus\boldsymbol{a})}[\log P(\boldsymbol{a}|\boldsymbol{\lambda})p(\boldsymbol{G}|\boldsymbol{a},\boldsymbol{K})]\} \\ \propto& \exp\left\{-\frac{1}{2}\left[\boldsymbol{a}^{\mathrm{T}}\left(\mathrm{diag}(\tilde{\boldsymbol{\lambda}})+\sum_{d=1}^{D}\sum_{m=1}^{P_d}(\boldsymbol{K}_{dm})_{\cdot\cdot}(\boldsymbol{K}_{dm})_{\cdot\cdot}^{\mathrm{T}}\right)\boldsymbol{a}\right.\right. \\ & -\boldsymbol{a}^{\mathrm{T}}\sum_{d=1}^{D}\sum_{m=1}^{P_d}(\boldsymbol{K}_{dm})_{\cdot\cdot}\widetilde{\boldsymbol{G}_{dm\cdot}^{\mathrm{T}}} \\ & \left.\left.-\sum_{d=1}^{D}\sum_{m=1}^{P_d}\widetilde{\boldsymbol{G}_{dm\cdot}}(\boldsymbol{K}_{dm})_{\cdot\cdot}^{\mathrm{T}}\boldsymbol{a}\right]\right\}. \end{aligned}$$

From the formula of $q(\boldsymbol{a})$ we know that the calculation process can update the mean and covariance matrix simultaneously which can reveal the correlation information of the samples. Through similar calculation, the approximate posterior distributions of $\boldsymbol{b}$ and $(e,\boldsymbol{c})$ can be obtained as

$$\begin{aligned} q(\boldsymbol{b}_{\cdot d}) \propto& \exp\left\{-\frac{1}{2}\left[\boldsymbol{b}_{\cdot d}^{\mathrm{T}}\left(\mathrm{diag}(\widetilde{\boldsymbol{\eta}_{\cdot d}})+\widetilde{\boldsymbol{G}_{d\cdot\cdot}\boldsymbol{G}_{d\cdot\cdot}^{\mathrm{T}}}\right)\boldsymbol{b}_{\cdot d}\right.\right. \\ & \left.\left.-\boldsymbol{b}_{\cdot d}^{\mathrm{T}}\widetilde{\boldsymbol{G}_{d\cdot\cdot}}\widetilde{\boldsymbol{L}_{d\cdot}^{\mathrm{T}}}-\widetilde{\boldsymbol{L}_{d\cdot}}\widetilde{\boldsymbol{G}_{d\cdot\cdot}^{\mathrm{T}}}\boldsymbol{b}_{\cdot d}\right]\right\}, \qquad d=1,...,D, \end{aligned}$$

and

$$\begin{aligned} q(e,\boldsymbol{c}) \propto& \exp\left\{-\frac{1}{2}\left[\begin{pmatrix}e\\ \boldsymbol{c}\end{pmatrix}^{\mathrm{T}}\right.\right. \\ & \begin{pmatrix}\tilde{\omega}+N & \mathbb{1}^{\mathrm{T}}\widetilde{\boldsymbol{L}^{\mathrm{T}}}\\ \tilde{\boldsymbol{L}}\mathbb{1} & \mathrm{diag}(\tilde{\boldsymbol{\gamma}})+\widetilde{\boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}}\end{pmatrix}\begin{pmatrix}e\\ \boldsymbol{c}\end{pmatrix} \\ & \left.\left.-\begin{pmatrix}e\\ \boldsymbol{c}\end{pmatrix}^{\mathrm{T}}\begin{pmatrix}\mathbb{1}^{\mathrm{T}}\tilde{\boldsymbol{f}}\\ \tilde{\boldsymbol{L}}\tilde{\boldsymbol{f}}\end{pmatrix}-\begin{pmatrix}\mathbb{1}^{\mathrm{T}}\tilde{\boldsymbol{f}}\\ \tilde{\boldsymbol{L}}\tilde{\boldsymbol{f}}\end{pmatrix}^{\mathrm{T}}\begin{pmatrix}e\\ \boldsymbol{c}\end{pmatrix}\right]\right\}. \end{aligned}$$

Third, the approximate posterior distributions of the intermediate outputs $\boldsymbol{G}$ and $\boldsymbol{L}$ are

$$\begin{aligned} q(\boldsymbol{G}) =& \prod_{d=1}^{D}\prod_{i=1}^{N}\mathcal{N}\left(\boldsymbol{G}_{d\cdot i};\boldsymbol{\Sigma}(\boldsymbol{G}_{d\cdot i})\left(\begin{pmatrix}(\boldsymbol{K}_{d1})_{i\cdot}\\ \vdots\\ (\boldsymbol{K}_{dP_d})_{i\cdot}\end{pmatrix}\tilde{\boldsymbol{a}}+\widetilde{\boldsymbol{b}_{\cdot d}}\widetilde{L_{di}}\right),\right. \\ & \left.\left(\boldsymbol{I}+\widetilde{\boldsymbol{b}_{\cdot d}\boldsymbol{b}_{\cdot d}^{\mathrm{T}}}\right)^{-1}\right), \end{aligned}$$

$$q(\boldsymbol{L}) = \prod_{i=1}^{N}\mathcal{N}\left(\boldsymbol{L}_{\cdot i};\boldsymbol{\Sigma}(\boldsymbol{L}_{\cdot i})(\tilde{\mathcal{G}}+\tilde{\boldsymbol{c}}\tilde{f}_i-\widetilde{\boldsymbol{c}e}),\left(\boldsymbol{I}+\widetilde{\boldsymbol{c}\boldsymbol{c}^{\mathrm{T}}}\right)^{-1}\right).$$

Through the update of the parameters above, we obtain the following approximate posterior distribution of $\boldsymbol{f}$.

$$q(\boldsymbol{f}) = \prod_{i=1}^{N}\mathcal{TN}\left(f_i;\widetilde{\boldsymbol{c}^{\mathrm{T}}}\widetilde{\boldsymbol{L}_{\cdot i}}+\tilde{e},1,f_i y_i>\tau\right). \qquad \square$$

Under the framework of approximate inference [6], the convergence criterion of the algorithm is summarized in Proposition 2.

Proposition 2. *The update of parameters converges by achieving the local maximum value of the lower bound as below.*

$$\begin{aligned} &\log p(\boldsymbol{y}|\boldsymbol{K}) \\ &\quad \geq \mathrm{E}_{q(\boldsymbol{\Theta},\boldsymbol{\Phi})}[\log p(\boldsymbol{y},\boldsymbol{\Theta},\boldsymbol{\Phi}|\boldsymbol{K})]-\mathrm{E}_{q(\boldsymbol{\Theta},\boldsymbol{\Phi})}[\log q(\boldsymbol{\Theta},\boldsymbol{\Phi})]. \end{aligned}$$

Algorithm 1 summarizes the calculation process of updating approximate posterior distributions.

3.2 Classification

Given a new data point $\boldsymbol{x}_*$, denote the kernel matrix set as $\boldsymbol{K}^*$. According to the training parameters, the predictive distribution of the first layer $\boldsymbol{G}_{d\cdot *}$ of intermediate output can be obtained by replacing the posterior distribution of $\boldsymbol{a}$ with its approximation $q(\boldsymbol{a})$ as follows:

$$P(\boldsymbol{G}_{d\cdot *}|\boldsymbol{K}^*,\boldsymbol{a},\boldsymbol{K},\boldsymbol{y}) = \prod_{m=1}^{P_d}\mathcal{N}\left(G_{dm*};\boldsymbol{\mu}(G_{dm*}),\boldsymbol{\Sigma}(G_{dm*})\right),$$

Algorithm 1 Calculation of approximate posterior distributions in BKAGL

Input: the kernel matrix set $\boldsymbol{K}$ and the label vector $\boldsymbol{y}$ of the training set, the hyper-parameter set $\boldsymbol{\Xi}$, the iteration number *iter*;

initial parameters of distributions of $\{\boldsymbol{\Theta}, \boldsymbol{\Phi}\}$;
for $t = 1, \ldots, iter$ **do**
  compute $\beta^t(\lambda_i)$ and related parameters;
  compute $\boldsymbol{\mu}^t(\boldsymbol{a})$, $\boldsymbol{\Sigma}^t(\boldsymbol{a})$ and related parameters;
  compute $\boldsymbol{\mu}^t(\boldsymbol{G}_{d\cdot i})$, $\boldsymbol{\Sigma}^t(\boldsymbol{G}_{d\cdot i})$ and related parameters;
  compute $\beta^t(\eta_{md})$ and related parameters;
  compute $\boldsymbol{\mu}^t(\boldsymbol{b}_{\cdot d})$, $\boldsymbol{\Sigma}^t(\boldsymbol{b}_{\cdot d})$ and related parameters;
  compute $\boldsymbol{\mu}^t(\boldsymbol{L}_{\cdot i})$, $\boldsymbol{\Sigma}^t(\boldsymbol{L}_{\cdot i})$ and related parameters;
  compute $\beta^t(\gamma_d)$ and related parameters;
  compute $\beta^t(\omega)$ and related parameters;
  compute $\boldsymbol{\mu}^t(e, \boldsymbol{c})$, $\boldsymbol{\Sigma}^t(e, \boldsymbol{c})$ and related parameters;
  compute $\boldsymbol{\mu}^t(\boldsymbol{f})$ and related parameters;
end for

Output: parameters of the approximate posterior distributions of $\{\boldsymbol{\Theta}, \boldsymbol{\Phi}\}$.

where

$$\boldsymbol{\mu}(G_{dm*}) = \boldsymbol{\mu}^{\mathrm{T}}(\boldsymbol{a})(\boldsymbol{K}_{dm})_{\cdot *}$$

and

$$\boldsymbol{\Sigma}(G_{dm*}) = 1 + (\boldsymbol{K}_{dm})_{\cdot *}^{\mathrm{T}} \boldsymbol{\Sigma}(\boldsymbol{a})(\boldsymbol{K}_{dm})_{\cdot *}.$$

Similarly, the predictive distribution of the second layer L_{d*} of intermediate output is

$$\begin{aligned} &P(L_{d*}|\boldsymbol{G}_{d\cdot *}, \boldsymbol{b}_{\cdot d}) \\ &\quad = \mathcal{N}(L_{d*}; \boldsymbol{\mu}^{\mathrm{T}}(\boldsymbol{b}_{\cdot d})\boldsymbol{G}_{d\cdot *}, 1 + \boldsymbol{G}_{d\cdot *}^{\mathrm{T}} \boldsymbol{\Sigma}(\boldsymbol{b}_{\cdot d})\boldsymbol{G}_{d\cdot *}). \end{aligned}$$

Then the predictive distribution of f_* can be formulated as

$$P(f_*|\boldsymbol{L}_{\cdot *}, \boldsymbol{c}, e) = \mathcal{N}\left(f_*; \boldsymbol{\mu}(f_*), \boldsymbol{\Sigma}(f_*)\right),$$

where

$$\boldsymbol{\mu}(f_*) = \boldsymbol{\mu}^{\mathrm{T}}(e, \boldsymbol{c}) \begin{pmatrix} 1 \\ \boldsymbol{L}_{\cdot *} \end{pmatrix}$$

and

$$\boldsymbol{\Sigma}(f_*) = 1 + \begin{pmatrix} 1 & \boldsymbol{L}_{\cdot *}^{\mathrm{T}} \end{pmatrix} \boldsymbol{\Sigma}(e, \boldsymbol{c}) \begin{pmatrix} 1 \\ \boldsymbol{L}_{\cdot *} \end{pmatrix}.$$

Moreover, given the margin parameter τ, we can obtain the predictive distribution of label y_* as given in Proposition 3.

Proposition 3. *The predictive distribution of y_* for $\boldsymbol{x}_*$ is*

$$P(y_* = +1|\boldsymbol{K}^*, \boldsymbol{K}, \boldsymbol{y}) = Z_*^{-1} \Phi\left(\frac{\boldsymbol{\mu}(f_*) - \tau}{\boldsymbol{\Sigma}(f_*)}\right),$$

$$P(y_* = -1|\boldsymbol{K}^*, \boldsymbol{K}, \boldsymbol{y}) = Z_*^{-1} \Phi\left(\frac{-\boldsymbol{\mu}(f_*) - \tau}{\boldsymbol{\Sigma}(f_*)}\right),$$

Algorithm 2 Prediction of the label y_* for a new point $\boldsymbol{x}_*$

Input: the kernel vector set $\boldsymbol{K}^*$ of the new point, the training parameters obtained from Algorithm 1;

compute $p(\boldsymbol{G}_{d\cdot *}|\boldsymbol{K}^*, \boldsymbol{a}, \boldsymbol{K}, \boldsymbol{y})$;
compute $p(L_{d*}|\boldsymbol{G}_{d\cdot *}, \boldsymbol{b})$;
compute $p(f_*|\boldsymbol{L}_{\cdot *}, \boldsymbol{c}, e)$;
compute $p(y_* = +1|\boldsymbol{K}^*, \boldsymbol{K}, \boldsymbol{y})$ and $p(y_* = -1|\boldsymbol{K}^*, \boldsymbol{K}, \boldsymbol{y})$;
if $p(y_* = +1|\boldsymbol{K}^*, \boldsymbol{K}, \boldsymbol{y}) \geq 0.5$ **then**
  label $= +1$;
else
  label $= -1$;
end if

Output: the label of $\boldsymbol{x}_*$.

where $Z_* = \Phi(\frac{\mu(f_*)-\tau}{\Sigma(f_*)}) + \Phi(\frac{-\mu(f_*)-\tau}{\Sigma(f_*)})$ is the normalization coefficient and $\Phi(\cdot)$ represents the cumulative distribution function of the standard normal distribution.

According to Proposition 3, we can calculate the classification probability of the test instance by Algorithm 2. Note that Algorithm 2 distinguishes the label by 0.5. That means 0.5 acts as the cut point. In fact, other cut points can also be selected through cross-validation techniques if needed.

4. EXPERIMENTS

In this section, we provide experiments on simulated data and real data to illustrate the performance of the proposed BKAGL method. We compare the proposed method with the k-nearest neighbors (KNN) algorithm, support vector machine (SVM) algorithm, the generalized linear model (GLM) with lasso, random forest and naive Bayes. For these contrastive classification methods, matrix data should be vectorized into a vector pattern before classification. In other words, a $D \times R$ matrix should be transformed into a DR-dimensional vector for each instance. Select $k = 4$ for KNN through cross-validation. The generalized linear model we use is the logistic regression and we use k-fold cross-validation for GLM with lasso. With regard to BKAGL, apply the k-medoids algorithm for grouping. To reflect the advantage of grouping, we also compare the proposed method with BKAGL without grouping. For all the kernel selection problems involved in this section, such as SVM, BKAGL and BKAGL without grouping, we choose the linear kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{x}_j$ uniformly for a fair comparison. Other kernels, including the polynomial kernel and the Gaussian kernel, can be used, but parameter tuning is computationally expensive. Hence, we choose the linear kernel. Through the sensitivity analysis of the hyper-parameters, we find that the proposed method is not sensitive to the choice of the hyper-parameters. Therefore, we set the hyper-parameters $(\alpha_\lambda, \beta_\lambda, \alpha_\eta, \beta_\eta, \alpha_\gamma, \beta_\gamma, \alpha_\omega, \beta_\omega) = (1, 1, 1, 1, 1, 1, 1, 1)$ throughout the experiment study. We implement the proposed BKAGL method in R and the codes are available at https://github.com/wangxz021/bkagl.

4.1 Simulated data

In this part, we discuss the classification of two types of multi-dimensional datasets. We first introduce the generation models, and then give the comparison of the classification results. Finally, we analyze the interpretation of grouping and the learning parameters $\boldsymbol{c}$ and $\boldsymbol{b}$.

Consider the classification problem of the following two models. Each data point $\boldsymbol{X}$ has D dimensions and each dimension is a vector with R features.

- Model 1: the label $Y = +1$, $\boldsymbol{X}$ is a $D \times R$ matrix: $X_{dr} \sim \mathcal{N}(0,1), 1 \le d \le D, 1 \le r \le R$.
- Model 2: the label $Y = -1$, $\boldsymbol{X}$ is a $D \times R$ matrix: $X_{1r} \sim \mathcal{N}(h,1), (1-p)R+1 \le r \le R, 0 \le p \le 1$; $X_{dr} \sim \mathcal{N}(0,1), otherwise$.

From the construction, the two generation models only differ in the last pR features of the first dimension. The parameters p and h decide the difference between the two models. The parameter p denotes the proportion of the difference and h determines the size of the difference. When p and h are large, classification is obviously easier. On the contrary, small p and h result in smaller differences, which makes classification more difficult. On the other hand, the classification method needs to be sensitive to the fluctuation due to the small sample size.

Generate N instances, among which half of them are from the first model and the rest are from the second model. In the classification process, 80% of instances are selected randomly from all of the instances as the training set and the remaining 20% form the test set. The probability of successful classification in the test set is used to measure the performance of each method. The results reported are the average of 100 independent runs.

First, take $N = 100$, $D = 2$, $R = 100$ and fix $p = 50\%$, then compare the seven algorithms on different h chosen from $\{0.1 \times i, 3 \le i \le 10\}$. For the sake of comparable performance measure, the classification accuracy of every method is calculated on the same training set and test set. Repeat the computing process and average the results. Figure 3 displays the relation between the classification accuracies of the seven methods and the difference parameter h. To clearly show the results and the trend, the X-axis is reversed. From Figure 3, one can see that BKAGL has the best classification performance. When the difference parameter h is large, BKAGL has higher classification accuracies than GLM with lasso, random forest and KNN. When h becomes smaller, naive Bayes, SVM and BKAGL without grouping exhibit a faster drop and the superiority of BKAGL becomes more obvious. Meanwhile, BKAGL has the advantage of interpretation ability provided by grouping and the learning parameters, which will be introduced in more detail later.

Then we consider the impact of the difference ratio p on the classification results. The value of p decides the proportion of differences. We fix $N = 100$, $D = 2$, $R = 100$,

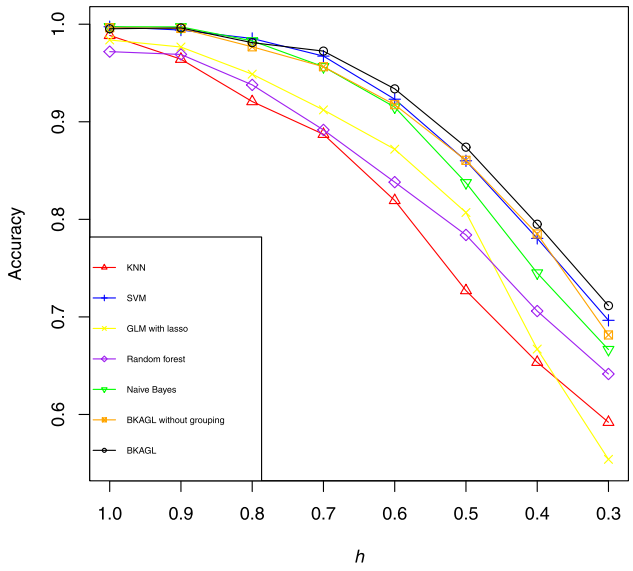


Figure 3. Classification accuracies of the seven methods under different h.

$h = 0.3$ and choose p from $\{10\%, 20\%, 30\%, 40\%, 50\%\}$. Repeat the calculation process and average the classification results. Table 1 shows the classification accuracies of the seven methods under different p. We observe from Table 1 that BKAGL performs constantly better than other methods under different p.

Next we discuss the learning parameters $\boldsymbol{c}$ and $\boldsymbol{b}$. Choose $N = 100$, $D = 2$, $R = 100$, $h = 1$, $p = 50\%$. Through observing the learning parameters in each iteration, we find that the grouping and the analysis results are consistent. Therefore, we randomly select one classification process from the iteration to provide the detailed analysis. The results are recorded in Table 2. We observe that $c_1 = 0.5029$ and $c_2 = 0.0000$. The value of $\boldsymbol{c}$ indicates that the classification result is only determined by the first dimension, which matches the simulation model. With regard to the learning parameter $\boldsymbol{b}$, whether all features are properly divided into several groups and the parameter $\boldsymbol{b}$ of different groups in different dimensions correctly reflects the importance of partial features are two important things that need to be verified. From the construction of the simulation model, we know that the key to classification is the second half features of the first dimension. Therefore, the features of the first dimension should become a group. The grouping and learning parameters in Table 2 provide the explanatory results of the classification.

The grouping results of the first dimension successfully select most of the second half features to form Group 2. Compared to the unimportant Group 1, the learning parameter $\boldsymbol{b}$ of Group 2 of the first dimension has a much higher value. The grouping of the second dimension looks random; nevertheless, the learning parameter $\boldsymbol{b}$ of each group is 0,

Table 1. Test accuracies of classification under different p

p	10%	20%	30%	40%	50%
KNN	0.5115±0.1148	0.5338±0.1166	0.5390±0.1102	0.5907±0.0969	0.5920±0.1091
SVM	0.5310±0.1114	0.5981±0.1126	0.5910±0.0960	0.6648±0.0993	0.6965±0.1067
GLM with lasso	0.4635±0.1000	0.4644±0.1004	0.4830±0.1209	0.5456±0.1396	0.5540±0.1263
Random forest	0.5155±0.1275	0.5513±0.1046	0.5690±0.1134	0.6011±0.1103	0.6415±0.1078
Naive Bayes	0.5230±0.1401	0.5438±0.1192	0.5685±0.0926	0.6126±0.1297	0.6665±0.1113
BKAGL without grouping	0.5345±0.1121	0.5894±0.0986	0.6060±0.1018	0.6670±0.0981	0.6815±0.1012
BKAGL	0.5380±0.1146	0.6156±0.1174	0.6325±0.1106	0.7033±0.0933	0.7115±0.1054

Table 2. The grouping of features and learning parameters

d	c_d	$P_d = 2$	*Real group*	*Experimental group*	b_{md}
1	0.5029	Group 1	1-50	1-18,20-28,30-32,34-50,52,67,78,98	0.1339
		Group 2	51-100	19,29,33,51,53-66,68-77,79-97,99-100	0.4288
d	c_d	$P_d = 2$	*Real group*	*Experimental group*	b_{md}
2	0.0000	Group 1	None	1,4,8,11,12,13,14,16,17,18,20,21,22,24,25,28,29,30,32, 33,34,36,37,38,39,41,42,45,52,53,54,57,59,60,63,65,67, 70,71,72,74,75,77,78,80,82,85,87,88,89,94,95,97,98,99,100	0.0000
		Group 2	None	2,3,5,6,7,9,10,15,19,23,26,27,31,35,40,43,44,46,47,48, 49,50,51,55,56,58,61,62,64,66,68,69,73,76,79,81,83,84, 86,90,91,92,93,96	0.0000

which shows the lack of importance on the final classification result.

The grouping and the learning parameters $\boldsymbol{c}$ and $\boldsymbol{b}$ accurately capture the characteristics of the simulation model. For practical applications, valuable information can be obtained from the grouping and the learning parameters.

4.2 Real data

In this section, we compare the performance of BKAGL with classical classification methods on two real datasets: ECG and EEG. Meanwhile, we will analyze the learning parameters $\boldsymbol{b}$ and $\boldsymbol{c}$ based on the practical meaning. The results reported are the average of 10 independent runs.

4.2.1 ECG dataset

The electrocardiogram (ECG) dataset [4] contains 200 instances, where 67 instances are abnormal (label +1) and 133 instances are normal (label −1). The researcher collected heartbeat data through two electrodes. Therefore, the heartbeat record for each instance is a multivariate time series. For instances in the raw data, the time series of each electrode have unequal lengths. To solve this problem, we choose to linearly interpolate the time series of each electrode so that every time series can have the same length as the longest length in the raw data. This is a common tool for data pre-processing [3, 9]. Some researchers used this form of pre-processing to show that for time series classification problems, the length of the time series is not an issue [23]. After the procedure, the length of the time series of each electrode for every instance is 152.

Table 3. Test accuracies of the ECG dataset

Method	Accuracy
KNN	0.6563±0.0182
SVM	0.7944±0.0474
GLM with lasso	0.6806±0.0526
Random forest	0.7456±0.0405
Naive Bayes	0.7150±0.0928
BKAGL without grouping	0.7738±0.0340
BKAGL	0.8056±0.0387

We note that the dimensions in the model in Section 2.2 are two electrodes and the features are 152 time points in this problem. We randomly select 40 instances as the training set and the remaining 160 instances form the test set. The classification accuracy of each method is shown in Table 3. As we can see, the proposed BKAGL outperforms other classification methods.

Next, we observe the learning parameters $\boldsymbol{c}$ and $\boldsymbol{b}$. We find that the grouping and analysis results are consistent under each iteration by observing each group of learning parameters. Consequently, we randomly select one process to give further analysis and the results are recorded in Table 4. The learning parameters c_d for the two dimensions (electrodes) are 0.4755 and 0.0000, respectively, which illustrates that the records on the first electrode have the most influence on the final output. On the other hand, Table 4 records the learning parameter $\boldsymbol{b}$ for the two dimensions. For each dimension, 152 time points are divided into two groups: earlier-stage (about the first two-thirds of time points) and later-stage (about the last third of time points). For the

first dimension, the earlier-stage period has a much higher absolute value than the later-stage period. For the second dimension, the learning parameters b_{m2} of two groups are very close to 0. According to the analysis based on the learning parameters $\boldsymbol{c}$ and $\boldsymbol{b}$, we know that the earlier-stage records of the first electrode play the most important role in the ECG classification problem.

4.2.2 EEG dataset

Next we consider the multiple electrode time series electroencephalography (EEG) dataset [8] of alcoholism (http://kdd.ics.uci.edu/databases/eeg/eeg.data.html). The study consists of 77 alcoholic subjects and 45 control subjects and contains voltage values which are measured from 64 electrodes placed on each subject's scalp. The 64 electrodes were located at standard sites (Standard Electrode Position Nomenclature, American Electroencephalographic Association 1990) [28]. The voltage values are recorded at 256 time points. Every subject had 120 trials exposed to three types of stimuli, which are single stimulus, two matched stimuli and two unmatched stimuli. The goal of the study is to explore the associations between alcoholism and voltage values over both channels and time.

We focus on the data exposed to the single stimulus and average all the trials under the single stimulus for each subject [17]. For intuitive cognition of the dataset, we average the voltage values of all alcoholic and control subjects respectively. Figure 4 displays the average EEG recordings of two groups. From Figure 4, we can see that the differences of two groups mainly lie in the medium-term and final performance of the EEG readings. Consider the medium-term and final data to find the relation among alcoholism, channels and the medium-term and final performance of the EEG readings.

Now the dimensions in the model in Section 2.2 are replaced by the 64 channels and the features are time points here. In other words, there are 64 time series data for each subject. According to the above analysis of Figure 4, the differences of two groups are mainly concentrated in the medium-term and final data. Hence, choose the 51st to 70th time points together with the 237th to 256th time points to be a time chain with length 40. Randomly select 100 subjects as the training set and the remaining 22 subjects form the test set. Calculate the classification accuracy of each method. Repeat the process and average the results. The classification results are shown in Table 5.

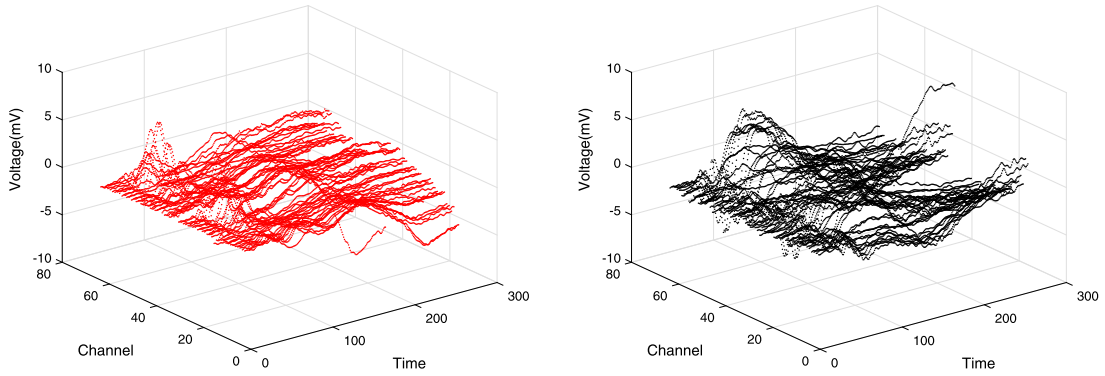
From Table 5 we know that BKAGL shows the best performance. Next we discuss the learning parameters $\boldsymbol{c}$ and $\boldsymbol{b}$ to find useful information. By observing each group of learning parameters under each iteration, we find the consistency of grouping and analysis results. Hence, randomly select one classification process from the iteration and give

Table 4. The grouping and learning parameters of the ECG dataset

d	c_d	$P_d = 2$	*Time points*	b_{md}
1	0.4755	Group 1	1-99	−0.4115
		Group 2	100-152	−0.0013
d	c_d	$P_d = 2$	*Time points*	b_{md}
2	0.0000	Group 1	1-100	0.0000
		Group 2	101-152	0.0000

Table 5. Test accuracies of the EEG dataset

Method	Accuracy
KNN	0.6705±0.0776
SVM	0.7159±0.0227
GLM with lasso	0.6705±0.0937
Random forest	0.7386±0.0435
Naive Bayes	0.6591±0.0587
BKAGL without grouping	0.7159±0.0937
BKAGL	0.7614±0.0435



(a) The alcoholic group (b) The control group

Figure 4. The average EEG recordings of two groups.

Table 6. The grouping and learning parameters of the EEG dataset

d	Channel name	c_d	$P_d = 3$	Time points	Period	b_{md}
55	PO7	1.1025	Group 1	51,52,53,54,55,56,57,58	Medium-term	0.2266
			Group 2	59,60,61,62,63,64,65,66,67,68,69,70	Medium-term	0.1974
			Group 3	237,238,239,240,241,242,243,244,245,246, 247,248,249,250,251,252,253,254,255,256	Final-stage	0.1794

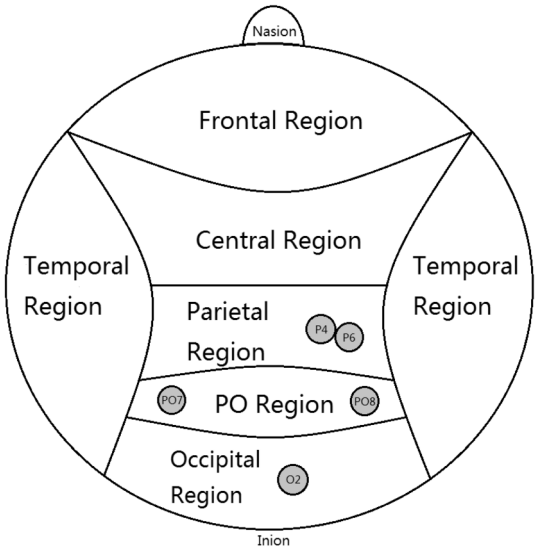


Figure 5. The locations of the top five channels on the scalp.

the following analysis. We obtain 64 c values for all channels. Different channels have different degrees of influence on the classification results. The order of the top five channels from high to low are PO7, P6, PO8, P4 and O2. According to electrode labelling, every placement site has a letter to identify the region of the brain. For example, P stands for Parietal and O stands for Occipital. PO means the intermediate electrode place between parietal and occipital. Odd and even numbers mean the left hemisphere and the right hemisphere, respectively. Each numeric value means a distance. Take PO7 as an example, number 7 is utilized for the left hemisphere to stand for 40% of the inion-to-nasion distance. We can observe from Figure 5 that the locations of the top five channels marked as grey circles are concentrated in the parietal region, occipital region and their intermediate region PO. Hence, the EEG readings of these regions are the key indices to classify the alcoholic group and the control group.

Finally, we discuss the learning parameter $\boldsymbol{b}$. We note that the 55th channel PO7 has the significantly largest c value that equals 1.1025, which means PO7 has the greatest impact on the classification. Therefore, we consider the learning parameter $\boldsymbol{b}$ of the 55th channel PO7, which is shown in Table 6. All 40 time points are divided into three groups where the 20 medium-term data points are split into two groups and the 20 final data points become a group independently. The parameters $\boldsymbol{b}$ of the two medium-term groups have larger values than that of the final-stage group, which means greater impacts on the final classification. More specifically, the parameter $\boldsymbol{b}$ of Group 1 has the largest value, which implies the importance of the time points in this group. On the whole, the medium-term of EEG readings has a higher influence on classification than the final performance of the EEG readings, which coincides with the rough observation of Figure 4.

The grouping and the learning parameters $\boldsymbol{c}$ and $\boldsymbol{b}$ can provide valuable information for practical applications. The proposed method BKAGL can be applied to various practical classification problems. The classification process is similar to the above analysis.

5. DISCUSSION AND EXTENSIONS

This paper proposes a novel kernel-based algorithm to solve the binary classification problem for multi-dimensional datasets. The proposed method can integrate information from different groups in different dimensions. Through the estimation of parameters, BKAGL has good interpretative ability, which enables us to find the groups of features and dimensions that have a greater impact on the response. Meanwhile, the utilization of variational approximation inference and conjugate Bayesian models provides the closed-form iterative formulas. For the multiclass classification problem, methods such as one-to-all [18] can be adopted. In addition, the correlation information between dimensions can be considered. In fact, if there is a priori knowledge about the dimensions, we can take advantage of it by designing a more reasonable prior distribution to improve the classification accuracy.

Received 1 August 2018

REFERENCES

[1] Altiparmak, F., Ferhatosmanoglu, H., Erdal, S. and Trost, D. C. (2006). Information mining over heterogeneous and high-dimensional time-series data in clinical trials databases. *IEEE Transactions on Information Technology in Biomedicine*, **10**(2), 254–263.

[2] Aubert, G. and Kornprobst, P. (2006). *Mathematical problems in image processing: partial differential equations and the calculus of variations.* Springer Science & Business Media, New York. MR2244145

[3] Bagnall, A., Lines, J., Bostrom, A., Large, J. and Keogh, E. (2017). The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, **31**(3), 606–660. MR3640659
[4] Baydogan, M. G. (2015). Multivariate time series classification datasets. http://www.mustafabaydogan.com.
[5] Beal, M. J. (2003). Variational algorithms for approximate Bayesian inference. PhD thesis, The Gatsby Computational Neuroscience Unit, University College London, London, The UK.
[6] Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer, New York. MR2247587
[7] Camacho, D. M., Collins, K. M., Powers, R. K., Costello, J. C. and Collins, J. J. (2018). Next-generation machine learning for biological networks. *Cell*, **173**(7), 1581–1592.
[8] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
[9] Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L. and Muller, P. A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, **33**(4), 917–963. MR3962039
[10] Ferreira, J. C. and Menegatto, V. A. (2009). Eigenvalues of integral operators defined by smooth positive definite kernels. *Integral Equations & Operator Theory*, **64**(1), 61–81. MR2501172
[11] Gönen, M. (2012). Bayesian efficient multiple kernel learning. *Proceedings of the 29th International Conference on Machine Learning*, Edinburgh, Scotland, The UK.
[12] Gönen, M. and Alpaydın, E. (2011). Multiple kernel learning algorithms. *Journal of Machine Learning Research*, **12**, 2211–2268. MR2825425
[13] Hastie, T., Tibshirani, R. and Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction, second edition*. Springer, New York. MR2722294
[14] Hofmann, T., Schölkopf, B. and Smola, A. J. (2008). Kernel methods in machine learning. *The Annals of Statistics*, **36**(3), 1171–1220. MR2418654
[15] Lawrence, N. D. and Jordan, M. I. (2005). Semi-supervised learning via Gaussian processes. *Advances in Neural Information Processing Systems*, **17**, 753–760.
[16] Lecun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning. *Nature*, **521**, 436–444.
[17] Li, B., Kim, M. K. and Altman, N. (2010). On dimension folding of matrix-or array-valued statistical objects. *The Annals of Statistics*, **38**(2), 1094–1121. MR2604706
[18] Liu, Y. and Zheng, Y. F. (2005). One-against-all multi-class SVM classification using reliability measures. *2005 IEEE International Joint Conference on Neural Networks*, **2**, 849–854.
[19] Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT Press, Cambridge, MA.
[20] Parisi, G. (1988). *Statistical field theory*. Addison-Wesley, Redwood City.
[21] Park, T. and Casella, G. (2008). The Bayesian lasso. *Journal of the American Statistical Association*, **103**(482), 681–686. MR2524001
[22] Peng, J., Zhu, X., Wang, Y., An, L. and Shen, D. (2019). Structured sparsity regularized multiple kernel learning for Alzheimer's disease diagnosis. *Pattern Recognition*, **88**, 370–382.
[23] Ratanamahatana, C. A. and Keogh, E. (2005). Three myths about dynamic time warping data mining. *Proceedings of the 2005 SIAM International Conference on Data Mining*, 506–510.
[24] Sonnenburg, S., Rätsch, G., Schäfer, C. and Schölkopf, B. (2006). Large scale multiple kernel learning. *Journal of Machine Learning Research*, **7**, 1531–1565. MR2274416
[25] Tipping, M. E. (2000). The relevance vector machine. *Advances in Neural Information Processing Systems*, **12**, 652–658.
[26] Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, **1**, 211–244. MR1875838
[27] Vapnik, V. N. (2000). *The nature of statistical learning theory, second edition*. Springer-Verlag, New York. MR1719582
[28] Zhang, X. L., Begleiter, H., Porjesz, B., Wang, W. and Litke, A. (1995). Event related potentials during object recognition tasks. *Brain Research Bulletin*, **38**(6), 531–538.
[29] Zhu, B., Song, N., Shen, R., Arora, A., Machiela, M.J., Song, L., Landi, M.T., Ghosh, D., Chatterjee, N., Baladandayuthapani, V. and Zhao, H. (2017). Integrating clinical and multiple omics data for prognostic assessment across human cancers. *Scientific Reports*, **7**(1), 1–13.

Xiaozhou Wang
School of Mathematical Sciences
Shanghai Jiao Tong University
Shanghai 200240
China
E-mail address: wangxiaozhou@sjtu.edu.cn

Fangli Dong
School of Mathematical Sciences
SJTU-Yale Joint Center of Biostatistics and Data Science
Shanghai Jiao Tong University
Shanghai 200240
China
E-mail address: dongfl@sjtu.edu.cn