Neural Computing and Applications
https://doi.org/10.1007/s00521-022-07282-6

ORIGINAL ARTICLE

An explainable multi-sparsity multi-kernel nonconvex optimization least-squares classifier method via ADMM

Zhiwang Zhang[1] · Jing He[2] · Jie Cao[1] · Shuqing Li[1] · Xingsen Li[3] · Kai Zhang[4] · Pingjiang Wang[4,5] · Yong Shi[6]

Received: 19 August 2021 / Accepted: 7 April 2022

Abstract

Convex optimization techniques are extensively applied to various models, algorithms, and applications of machine learning and data mining. For optimization-based classification methods, the sparsity principle can greatly help to select simple classifier models, while the single- and multi-kernel methods can effectively address nonlinearly separable problems. However, the limited sparsity and kernel methods hinder the improvement of the predictive accuracy, efficiency, iterative update, and interpretable classification model. In this paper, we propose a new Explainable Multi-sparsity Multi-kernel Nonconvex Optimization Least-squares Classifier (EM^{2}NOLC) model, which is an optimization problem with a least-squares objective function and multi-sparsity multi-kernel nonconvex constraints, aiming to address the aforementioned issues. Based on reconstructed multiple kernel learning (MKL), the proposed model can extract important instances and features by finding the sparse coefficient and kernel weight vectors, which are used to compute importance or contribution to classification and obtain the explainable prediction. The corresponding EM^{2}NOLC algorithm is implemented with the Alternating Direction Method of Multipliers (ADMM) method. On the real classification datasets, compared with the three ADMM classifiers of Linear Support Vector Machine Classifier, SVMC, Least Absolute Shrinkage and Selection Operator Classifier, the two MKL classifiers of SimpleMKL and EasyMKL, and the gradient descent classifier of Feature Selection for SVMC, our proposed EM^{2}NOLC generally obtains the best predictive performance and explainable results with the least number of important instances and features having different contribution percentages.

Keywords Least squares · Multiple kernel learning · Sparse learning · Explainable · Nonconvex optimization · Classification

✉ Zhiwang Zhang
9120211050@nufe.edu.cn

[1] College of Information Engineering, Nanjing University of Finance and Economics, Nanjing 210023, China

[2] Department of Neuroscience, University of Oxford, Oxford, UK

[3] Research Institute of Extenics and Innovation Methods, Guangdong University of Technology, Guangzhou 510006, China

[4] Quanzhou HUST Research Institute of Intelligent Manufacturing, Quanzhou 362000, China

[5] School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

[6] Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing 100190, China

Published online: 03 May 2022

1 Introduction

Many problems of machine learning and data mining are formulated as optimization models, which are often represented as minimization or maximization problems of objective functions subject to constraints on variables or parameters. Conversely, optimization techniques facilitate the process of solving machine learning problems by finding their solutions in an efficient way [1]. Classification is one of the main tasks in data mining and inductive learning methods, and it maps input data to a discrete set by fitting a model which is usually called a classifier. Most classification problems can be defined as optimization models, and they are solved by optimization algorithms [2, 3].

Over more than two decades, Support Vector Machines (SVM), as the fruit of statistical learning theory and optimization, have increased in popularity due to high predictive accuracy on the small- and medium-scale data [4–7]. The main idea of the SVM Classifier (SVMC) is to find an optimal decision hyperplane separating two classes by using a regularization parameter to reach a trade-off between the total error and the model complexity, which is defined as the squared ℓ_2 - norm of the weight vector [4, 6]. Then a decision function based on a small number of support vectors is constructed to predict the class labels of unseen input points. The primal SVMC model is an effective classifier for linearly separable problems, so it is also called the Linear SVM Classifier (LSVMC) [5]. Although the primal SVMC model can obtain feature weights, it may give a poor predictive performance for nonlinearly separable data. The dual SVMC model can get instance coefficients, whereas it is unable to explicitly compute feature weights. Therefore, SVMC can obtain the limited sparsity of instance coefficients, but it has no ability for feature selection or dimensionality reduction.

Introducing kernel functions into classifier models can effectively address nonlinearly separable problems [8], i.e. the use of single- and multi-kernel functions in classifier models can help to sufficiently learn the distinction of different classes. In particular, Multiple Kernel Learning (MKL) remarkably improves predictive accuracy of classifiers, where multiple kernel functions are combined with corresponding parameters in different ways [9, 10]. Thus, many MKL methods are proposed, including linear SimpleMKL [9], nonlinear MKL [11], multi-class MKL [12], two-stage MKL [13, 14], large-scale MKL [15], scalable EasyMKL [16], deep learning MKL [17], and sparse MKL [18] approaches. However, when kernel functions are integrated into the SVMC model, we obtain not feature weights but instance coefficients. Besides, training multi-kernel classifiers consumes a lot of system resources so their computational efficiency is low.

To obtain sparse solutions of optimization problems, some sparsity-inducing norms regarding variables are added to the SVMC model with the aim of feature selection [19–21]. The ℓ_1- and ℓ_0 - norm regularization methods are often employed to control the model complexity and avoid overfitting, and their sparsity decreases as the order of two norms increases. The ℓ_0 - norm regularization can obtain the minimum number of variables in theory, but the corresponding mathematical problem is difficult to solve in practice, because it is nonconvex and discontinuous [22]. There are two methods to cope with this situation. One is that an approximation function is defined and then it is used to replace the ℓ_0 - norm for dimensionality reduction [18, 23, 24]. Another one is that ℓ_0 - norm is substituted for ℓ_1 - norm. Although ℓ_1 - norm is a non-smooth function, it is convex and can generate relatively sparse solutions [1, 25–28]. If the total error and the squared ℓ_2 - norm of weights in the SVMC model are, respectively, replaced with the error sum of squares and the ℓ_1 - norm of weights, then we obtain the Least Absolute Shrinkage and Selection Operator Classifier (LASSOC) [29–31]. The LASSOC method can select a feature subset by a sparse weight vector. However, it is unable to extract an instance subset, while it is very difficult to introduce kernel functions into LASSOC because of the absolute function in the ℓ_1 - norm regularization. Therefore, it has poor predictive performance for nonlinearly separable data.

Owing to the explosive growth in size and complexity of data, it is required us to solve optimization problems with a large number of instances or features. For the freedom from the impacts of noise and redundancy, the structural sparsity imposed on objective or constraint functions results in nonconvex optimization problems, which are often NP-hard to solve [22]. In order to accurately capture the characteristics of large-scale or high-dimensional data and get high predictive accuracy, some algorithms, for instance, Feature Selection for SVMC (FS-SVMC) using gradient descent (GD) and sparse feature coefficients integrated into linear and nonlinear kernel functions [32], proximal [33], parallel or distributed [34–37], heuristic or iterative [38, 39], and online [40, 41], are proposed to solve such optimization problems. However, these methods only serve a single purpose, such as sparsity, approximation, decentralization, iterative update, and online learning. As for sparsity, many classifier models aim to get either a sparse coefficient vector of instances or a sparse weight vector of features.

In recent years, deep learning has provided a systemic solution to large-scale machine learning problems, especially for multimedia data processing [42–44], such as image, video, speech, and text. However, due to the nonlinear mapping in neural network structures, it is difficult to obtain sparse solutions of instances or features, so deep learning has poor interpretability.

To make clear the process of model learning and prediction, interpretability plays an important role in many practical applications [45–47], such as medical prognosis, finance, and gene or protein expression analysis. For linear models and rule-based systems, it is easy to compute the importance of features by using their weights and intervals. However, for nonlinear models, it is difficult to directly obtain instance coefficients or feature weights. Therefore, some probabilistic or statistical methods are used to assist in generating explainable results on the importance of instances or features.

The main challenge of optimization-based classification is that noisy and redundant instances or features in data degrade the predictive performance of classifiers. Therefore, our main motivation is to construct new optimization classifier methods to find spare vectors of coefficients and weights, which determine whether an instance or a feature is useful or important to prediction or not, so as to enhance the predictive accuracy and interpretability of classification.

In this study, we propose a new two-stage Explainable Multi-sparsity Multi-kernel Nonconvex Optimization Least-squares Classifier (EM^{2}NOLC) model and the corresponding algorithm implemented in the framework of Alternating Direction Method of Multipliers (ADMM). The classifier approach overcomes the drawbacks of LSVMC, SVMC, LASSOC, SimpleMKL, EasyMKL, and FS-SVMC, i.e., it can simultaneously select important instances and relevant features to provide explainable prediction apart from classification.

Our main contributions include: (i) a non-separable function in constraints with respect to the coefficient and weight vectors are transformed into two separable parts. (ii) the computation of row-wise multi-kernel matrix makes it easier to extract important instances with nonzero coefficients. (iii) the calculation of column-wise multi-kernel matrix makes it more convenient to select important features with nonzero weights. (iv) our proposed EM^{2}NOLC is able to provide the interpretable classification by using the minimum number of important instances and features with various contribution percentages, apart from prediction. (v) the EM^{2}NOLC algorithm implemented in the ADMM framework can efficiently solve classification problems in an iterative way.

2 Related work

Due to the constant growth in scale and complexity of data, there are increasing requirements for solving problems with a large number of instances and features. For optimization problems in data mining and machine learning, ADMM is one of the effective methods to address these problems, especially for problems with separable objective and equality constraint functions. According to the description in the literatures [5, 6, 34, 36], some classifier models based on convex optimization, for example, LSVMC, SVMC, and LASSOC, and their ADMM algorithms are extensively used to solve optimization problems in various applications.

The LSVMC model can be expressed as an unconstrained convex optimization problem [6, 38]. The objective function is composed of the hinge loss function and the squared ℓ_2 - norm of weights, and the trade-off between them is achieved by a regularization parameter. The feature weights can be obtained from the solution of the LSVMC model, but the sparsity of weights is very limited. Since the hinge loss is a discontinuous and non-differentiable function, the LSVMC algorithm with ADMM has poor efficiency in the process of weight minimization.

The dual SVMC model is the convex quadratic optimization problem [33, 34, 36], which consists of quadratic objective function and equality constraints regarding Lagrange multipliers or coefficients of instances. Then the SVMC algorithm in the framework of ADMM can be denoted as a linear equation system by using the proximal operator of the coefficient vector of instances. Owing to the nonnegative constraint of equivalence variables of coefficients, the update step of coefficients converges to a sparse vector. Thus, introducing kernel functions into the SVMC algorithm with ADMM helps to produce the sparse coefficient vector of instances. However, the sparsity of coefficients is fairly limited and it is unable to give the weight vector of features.

The LASSOC model can also be formulated as an unconstrained convex optimization problem [33, 34, 36, 48]. A regularization parameter is used to get a balance between the least-squares loss and the ℓ_1-norm of weights. Since the soft thresholding operator is imposed on equivalence variables of weights, the minimization of the augmented Lagrange function regarding the least-squares loss generates the sparse weight vector of features by the Karush–Kuhn–Tucker optimality condition.

Different from kernel methods in the SVMC model and the MKL techniques in SimpleMKL [9] and EasyMKL [16], this paper reconstructs row- and column-wise multi-kernel matrices with different features in a bottom-up fashion. At the same time, the ℓ_0-norms of instance coefficients and feature kernel weights are added to the constraints of the EM^{2}NOLC model instead of the ℓ_0 - norm of feature coefficients in the FS-SVMC model [32], the ℓ_1-norm of weights in the LASSOC model, and the ℓ_2-norm of weights in the LSVMC and SVMC models. Therefore, the EM^{2}NOLC model is denoted as a multi-sparsity multi-kernel nonconvex optimization problem, and the EM^{2}NOLC algorithm via ADMM converges to highly sparse coefficient and weight vectors. Then the sparse coefficient and weight vectors are, respectively, employed to compute the importance and contribution of instances and features to classification in order to further obtain an explainable prediction.

The rest of this paper is organized as follows: Sect. 3 demonstrates the novel EM^{2}NOLC model and the corresponding algorithm with ADMM. The experimental results of seven classifiers on ten real datasets, the comparison analysis of predictive performance, the importance analysis of instances and features, and the correlation analysis of selected features across folds are presented in Sect. 4. Finally, discussion and conclusions are given in Sects. 5 and 6, respectively.

3 Our proposed EM^{2}NOLC approach

This section presents an elaboration of our proposed EM^{2}NOLC approach. The optimization model of EM^{2-}NOLC is firstly described, and the EM^{2}NOLC algorithm using ADMM is then given.

3.1 The EM^2 NOLC model

Since the least-squares method has the advantages in stability and robustness of solutions, it has been widely used to construct various classifier models [49–51]. However, the predictive performance of classifiers is vulnerable to the influence of high variance from noise and redundancy in data. Moreover, they are unable to obtain sparse solutions from complex and redundant data, and important instances and features are hardly found to improve predictive accuracy and interpretability. For these reasons, a new EM^{2}NOLC model is constructed, which can simultaneously select important instances and features from the training set in addition to classification.

Generally, a binary classification problem can be expressed as: given training data $\boldsymbol{T} = \{\ (\boldsymbol{x}_i, y_i)|i \in \mathbb{N}\}\ _{i=1}^{n}$ with an instance set $\boldsymbol{X} = \{\boldsymbol{x}_i|i \in \mathbb{N}\}_{i=1}^{n}(\boldsymbol{x}_i \in \mathbb{R}^d)$ and a feature set $\boldsymbol{F} = \{\boldsymbol{f}_m|m \in \mathbb{N}\}_{m=1}^{d}(\boldsymbol{f}_m \in \mathbb{R}^n)$, each input point or instance $\boldsymbol{x}_i(\boldsymbol{x}_i \in \mathbb{R}^d)$ belongs to either of the two classes with a label y_i $(y_i \in \{-1, 1\})$, where d is the dimensional size of the input space and n is the sample size. The primal least-squares optimization classifier problem with the ℓ_2 - norm of errors and equality constraints is defined as

$$\begin{aligned}&\min_{w,b,\xi} \frac{1}{2}\|\xi\|_2^2\\ &s.t.\, y_i\{\boldsymbol{w}^T\phi(\boldsymbol{x}_i) - b\} = 1 + \xi_i,\ \xi_i \in \mathbb{R},\ i = 1, \cdots, n.\end{aligned} \tag{1}$$

The weight vector $\boldsymbol{w}$ can be further expressed as the linear combination regarding the coefficient vector $\boldsymbol{\lambda}$ $(\boldsymbol{\lambda} \in \mathbb{R}^n)$ of instances, that is $\boldsymbol{w} = \sum_{j=1}^{n} \lambda_j \phi(\boldsymbol{x}_i)$. For any two input points $\boldsymbol{x}_i(i = 1, \cdots, n)$ and $\boldsymbol{x}_j(j = 1, \cdots, n)$ from the training data $\boldsymbol{T}$, the dot product $\phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)$ in a new feature space is substituted with the kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Hence, the constraint condition in the equality constrained optimization problem (1) is rewritten as

$$y_i\left\{\sum_{j=1}^{n} \lambda_j K(\boldsymbol{x}_j, \boldsymbol{x}_i) - b\right\} = 1 + \xi_i,\ \xi_i \in \mathbb{R},\ i = 1, \cdots, n \tag{2}$$

Based on the MKL methods [11–13], the above kernel function $K(\boldsymbol{x}_j, \boldsymbol{x}_i)$ can be denoted as the linear combination of multiple feature kernels regarding the kernel weights $\boldsymbol{\mu}(\boldsymbol{\mu} \in \mathbb{R}^d)$, that is $K(\boldsymbol{x}_j, \boldsymbol{x}_i) = \sum\limits_{m=1}^{d} \mu_m \kappa(\boldsymbol{x}_{jm}, \boldsymbol{x}_{im})$. So, if it is integrated into the constraint condition (2), then we have the new constraint function

$$\begin{aligned}&y_i\left\{\sum\nolimits_{j=1}^{n} \lambda_j \sum\nolimits_{m=1}^{d} \mu_m \kappa(\boldsymbol{x}_{jm}, \boldsymbol{x}_{im}) - b\right\} = 1 + \xi_i,\ \xi_i\\ &\in \mathbb{R},\ i = 1, \cdots, n\end{aligned} \tag{3}$$

If the coefficient λ_i and the weight μ_m are employed to indicate whether an instance or a feature is important or contribution to classification or not, then they are important for the case of $\lambda_i \neq 0$ or $\mu_m \neq 0$. Otherwise, they are unimportant, and can be removed from training set. The larger the values of λ_i and μ_m, the more important they are. For the sake of efficiency and interpretability, without loss of classifier performance, we hope that the smaller the number of $\lambda_i \neq 0$ and $\mu_m \neq 0$ the better. So, the primal optimization problem of the EM^{2}NOLC model has the form

$$\begin{aligned}&\min_{\boldsymbol{\lambda},\boldsymbol{\mu},b,\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{\xi}\|_2^2\\ &s.t.\, y_i\left\{\sum_{j=1}^{n} \lambda_j \sum_{m=1}^{d} \mu_m \kappa(\boldsymbol{x}_{jm}, \boldsymbol{x}_{im}) - b\right\} = 1 + \xi_i,\\ &\|\boldsymbol{\lambda}\|_0 \leq C_\lambda,\ \|\boldsymbol{\mu}\|_0 \leq C_{\boldsymbol{\mu}},\\ &\boldsymbol{\lambda} \in \mathbb{R}^n,\ \boldsymbol{\mu} \in \mathbb{R}^d,\ \xi_i \in \mathbb{R},\ i = 1, \cdots, n\end{aligned} \tag{4}$$

involving the user-expected maximum numbers $C_\lambda(C_\lambda \in \mathbb{N})$ of important instances and $C_\mu(C_\mu \in \mathbb{N})$ of important features, and the intercept $b(b \in \mathbb{R})$.

Let $F(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{j=1}^{n} \lambda_j \sum_{m=1}^{d} \boldsymbol{\mu}_m \boldsymbol{\kappa}(\boldsymbol{x}_{jm}, \boldsymbol{x}_{im})$, it is a nonseparable function because of its mutually dependent variables of the coefficients $\boldsymbol{\lambda}$ and the kernel weights $\boldsymbol{\mu}$. For any two input points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ from the training data $\boldsymbol{T}$, the radial basis kernel function (RBF) $\kappa(x_{im}, x_{jm})$ with respect to the mth feature $(m = 1, \cdots, d)$ is defined as

$$\kappa(x_{im}, x_{jm}) = \exp\left(-(x_{im} - x_{jm})^2 \Big/ 2\sigma^2\right) \tag{5}$$

where $exp(\cdot)$ is the exponential function, and the bandwidth σ $(\sigma > 0)$ is a user-specified parameter.

Suppose that the kernel weight vector $\boldsymbol{\mu}$ of features is given, then $F(\boldsymbol{\lambda}, \boldsymbol{\mu})$ is a function regarding the coefficients $\boldsymbol{\lambda}$. We define the matrix $\boldsymbol{A}(\boldsymbol{A} \in \mathbb{R}^{n \times n})$ as the row-wise multi-kernel matrix, which is computed by

$$\boldsymbol{A} = \sum_{m=1}^{d} \mu_m \kappa(\boldsymbol{x}_{jm}, \boldsymbol{x}_{im}),\ i, j = 1, \cdots, n \tag{6}$$

The proof that the equality (6) holds can be found in "Proof of equality (6)" in Appendix. Thus, from the problem (4) and Eq. (6) we obtain the nonconvex constrained optimization problem of the first-stage EM^{2}NOLC model, which is called the λ - step EM^{2}NOLC, and it has the form

$$\begin{aligned} \lambda \text{ - step}: \min_{\lambda, b_1} \frac{1}{2} \|\boldsymbol{y} \odot (\boldsymbol{A}\lambda - b_1 \mathbf{1}_n) - \mathbf{1}_n\|_2^2 \\ s.t.\ \|\boldsymbol{\lambda}\|_0 \le C_\lambda,\ \boldsymbol{\lambda} \in \mathbb{R}^n \end{aligned} \tag{7}$$

with the elementwise product $\odot$ of two vectors and the intercept $b_1 (b_1 \in \mathbb{R})$.

Similarly, there is another form $F(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{m=1}^{d} \mu_m \sum_{j=1}^{n} \lambda_j \kappa(\boldsymbol{x}_{jm}, \boldsymbol{x}_{im})$, given the coefficient vector $\boldsymbol{\lambda}$ of instances, then $F(\boldsymbol{\lambda}, \boldsymbol{\mu})$ is a function regarding the kernel weights $\boldsymbol{\mu}$. Now we define the matrix $\boldsymbol{B}(\boldsymbol{B} \in \mathbb{R}^{n \times d})$ as the column-wise multi-kernel matrix, which is computed by

$$\boldsymbol{B} = \sum_{j=1}^{n} \lambda_j \kappa(\boldsymbol{x}_{jm}, \boldsymbol{x}_{im}),\ i = 1, \cdots, n,\ m = 1, \cdots, d \tag{8}$$

The proof that the equality (8) is true can be found in "Proof of equality (8)" in "Appendix". Thus, from the problem (4) and Eq. (8) we obtain the nonconvex constrained optimization problem of the second-stage EM^{2}NOLC model, which is called the μ - step EM^{2}NOLC, and it has the form

$$\begin{aligned} \mu \text{ - step}: \min_{\mu, b_2} \frac{1}{2} \|\boldsymbol{y} \odot (\boldsymbol{B}\mu - b_2 \mathbf{1}_n) - \mathbf{1}_n\|_2^2 \\ s.t.\ \|\boldsymbol{\mu}\|_0 \le C_\mu,\ \boldsymbol{\mu} \in \mathbb{R}^d \end{aligned} \tag{9}$$

with the intercept $b_2 (b_2 \in \mathbb{R})$.

By this point the indecomposable function $F(\boldsymbol{\lambda}, \boldsymbol{\mu})$ has been transformed into two separable objective functions of the two-stage EM^{2}NOLC model. The optimization problems (7) and (9) of the $\lambda-$ and μ - step EM^{2}NOLC models have the convex objective functions and the nonconvex constraint sets. The constraint set in (7) is responsible for finding important instances with $\lambda_i \neq 0$ and the number of them is no more than a specified constant C_λ. The constrain condition in (9) is used for selecting important features with $\mu_m \neq 0$ and the number of them is no more than a predefined constant C_μ.

Under the condition of given the initial $\boldsymbol{\mu}$, the λ-step minimization of EM^{2}NOLC in (7) is firstly solved based on the row-wise multi-kernel matrix $\boldsymbol{A}$, and the new coefficient vector $\boldsymbol{\lambda}$ is obtained. Based on the new $\boldsymbol{\lambda}$ and column-wise multi-kernel matrix $\boldsymbol{B}$, then the μ-step minimization of EM^{2}NOLC in (9) is solved, and the new kernel weight vector $\boldsymbol{\mu}$ is gained. Therefore, on the basis of the new $\boldsymbol{\mu}$, the above $\lambda-$ and μ - step optimization problems are alternately minimized until the process is satisfied with the stop criterion

$$\|\boldsymbol{\mu}^t - \boldsymbol{\mu}^{t-1}\|_\infty \le \tau \tag{10}$$

where $\boldsymbol{\mu}^t$ and $\boldsymbol{\mu}^{t-1}$ are the kernel weight vectors at the adjacent two iterations t and $t-1$, and $\tau(\tau > 0)$ is a sufficiently small constant.

Once the stop condition in Eq. (10) is satisfied, we obtained the optimal coefficients $\overline{\lambda}$ and kernel weights $\overline{\mu}$. For $\overline{\lambda}_i \neq 0$, the input point $\boldsymbol{x}_i (i = 1, \cdots, n)$ is considered as an important point, which has a contribution to classification. Otherwise, the input point $\boldsymbol{x}_i$ with $\overline{\lambda}_i = 0$ may be a noise or redundant instance, and it can be omitted. The λ - step EM^{2}NOLC generates the sparse coefficient vector to extract important instances apart from prediction. Similarly, for $\overline{\mu}_m \neq 0$, the feature $\boldsymbol{f}_m (m = 1, \cdots, d)$ is regarded as an important feature for classification. Otherwise, the feature $\boldsymbol{f}_m$ with $\overline{\mu}_m = 0$ is redundant or unimportant, and it can be removed. The μ - step EM^{2}NOLC produces the sparse kernel weight vector to select important features in addition to classification.

Based on the optimal coefficients $\overline{\lambda}$ and kernel weights $\overline{\mu}$, the two intercepts b_1 and b_2 are obtained from.

$$b_1 = (1/n) \sum_{i=1}^{n} \boldsymbol{A}_i \overline{\lambda},$$

$$b_2 = (1/n) \sum_{i=1}^{n} \boldsymbol{B}_i \overline{\mu}$$

where the vectors $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$ are, respectively, the ith rows of the multi-kernel matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ corresponding with input point $\boldsymbol{x}_i$. So, for a new input point $\boldsymbol{x}$ from a test set, suppose that the vectors $\boldsymbol{A}_x$ and $\boldsymbol{B}_x$ are obtained, its class label can be predicted by one of the two decision functions

$$f_1(\boldsymbol{x}) = \text{sign}\left(\boldsymbol{A}_x\overline{\lambda} - b_1\right) \tag{11}$$

$$f_2(\boldsymbol{x}) = \text{sign}\left(\boldsymbol{B}_x\overline{\mu} - b_2\right) \tag{12}$$

In this paper, we employ the first decision function to give the prediction of class label of any input point.

3.2 The EM^{2}NOLC algorithm via ADMM

On the basis of the demonstrations of the $\lambda -$ and μ - step EM^{2}NOLC models in Sect. 3.1, we will give the EM^{2}NOLC algorithm in the ADMM framework. Since ADMM is one of the best methods to solve optimization problems with separable objective and equality constraint functions, we have to transform the optimization problems of the $\lambda -$ and μ - step EM^{2}NOLC models in (7) and (9) into the ADMM forms.

The λ - step EM^{2}NOLC model (7) can be formulated in ADMM form as follows:

$$\begin{aligned} &\min_{\boldsymbol{\lambda},\boldsymbol{q}\in\mathbb{R}^n} f(\boldsymbol{\lambda}) + g(\boldsymbol{q}) \\ &\quad s.t.\, \boldsymbol{\lambda} - \boldsymbol{q} = \mathbf{0}_n \end{aligned} \tag{13}$$

with $f(\boldsymbol{\lambda}) = (1/2)\|\boldsymbol{y} \odot (\boldsymbol{A}\lambda - b_1\mathbf{1}_n) - \mathbf{1}_n\|_2^2$ and $g(\boldsymbol{q}) = I_{S_0(C_\lambda)}(\boldsymbol{q})$, where $I_{S_0(C_\lambda)}(\boldsymbol{q})$ is the indicator function built on the nonconvex set of C_λ - sparse vectors, and we have $S_0(C_\lambda) = \{\boldsymbol{q} \in \mathbb{R}^n | \|\boldsymbol{q}\|_0 \le C_\lambda\}$.

For the optimization problem (13), we can formulate the λ-step EM^{2}NOLC algorithm in the scaled ADMM form as the iterative updates:

$$\boldsymbol{\lambda}^{k+1} = \arg\min_{\boldsymbol{\lambda}\in\mathbb{R}^n}\left\{f(\boldsymbol{\lambda}) + (\rho/2)\left\|\boldsymbol{\lambda} - \boldsymbol{q}^k + \boldsymbol{u}^k\right\|_2^2\right\} \tag{14}$$

$$\boldsymbol{q}^{k+1} = \arg\min_{\boldsymbol{q}\in\mathbb{R}^n}\left\{g(\boldsymbol{q}) + (\rho/2)\left\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{q} + \boldsymbol{u}^k\right\|_2^2\right\} \tag{15}$$

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k + \boldsymbol{\lambda}^{k+1} - \boldsymbol{q}^{k+1} \tag{16}$$

at iteration $k+1$, involving the scaled dual variables $\boldsymbol{u}(\boldsymbol{u} \in \mathbb{R}^n)$ and the penalty parameter $\rho(\rho > 0)$.

Let $\boldsymbol{A}_y = \boldsymbol{y} \odot \boldsymbol{A}$, we define $y \odot \boldsymbol{A} = (y \odot \boldsymbol{A}_1, \cdots, y \odot \boldsymbol{A}_n)$ and $\odot$ is the elementwise product between the label vector $\boldsymbol{y}$ and the row-wise multi-kernel vector $\boldsymbol{A}_j(j = 1, \ldots, n)$. By solving the KKT optimality condition of the proximal operator (14), the λ - update is written as

$$\boldsymbol{\lambda}^{k+1} = \left(\boldsymbol{A}_y^T\boldsymbol{A}_y + \rho\boldsymbol{I}_n\right)^{-1}\left\{\boldsymbol{A}_y^T(b_1\boldsymbol{y} + \mathbf{1}_n) + \rho\left(\boldsymbol{q}^k - \boldsymbol{u}^k\right)\right\} \tag{17}$$

Owing to $\rho > 0$, the matrix $\boldsymbol{A}_y^T\boldsymbol{A}_y + \rho\boldsymbol{I}_n$ is always invertible.

From the q - minimization(15), the q - update can be regarded as the orthogonal projection computation with the form

$$\boldsymbol{q}^{k+1} = \Pi_{S_0(C_\lambda)}(\boldsymbol{\lambda}^{k+1} + \boldsymbol{u}^k) \tag{18}$$

The proof for the derivation of the update formulas (16), (17), and (18) of the λ - step EM^{2}NOLC algorithm in the scaled ADMM form can be found in "Proof of the λ-step EM^{2}NOLC algorithm via ADMM" in Appendix.

According to the u - update(16), the primal residual of the ADMM problem (13) is computed by $\boldsymbol{r}_{\lambda-\text{step}}^{k+1} = \boldsymbol{\lambda}^{k+1} - \boldsymbol{q}^{k+1}$. The dual residual can be obtained from the augmented Lagrangian function of (13), and it is denoted as $\boldsymbol{s}_{\lambda-\text{step}}^{k+1} = \rho(\boldsymbol{q}^k - \boldsymbol{q}^{k+1})$. Thus, the stop criteria of the λ - step EM^{2}NOLC algorithm via ADMM are

$$\left\|\boldsymbol{r}_{\lambda-\text{step}}^k\right\|_2 \le \varepsilon_{primal} \text{ and } \left\|\boldsymbol{s}_{\lambda-\text{step}}^k\right\|_2 \le \varepsilon_{dual} \tag{19}$$

where $\varepsilon_{primal}(\varepsilon_{primal} > 0)$ and $\varepsilon_{dual}(\varepsilon_{dual} > 0)$ are, respectively, feasibility tolerances.

Similarly, in the ADMM form, the μ - step EM^{2}NOLC model (9) can be rewritten as

$$\begin{aligned} &\min_{\boldsymbol{\mu},\mathbf{z}\in\mathbb{R}^d} h(\boldsymbol{\mu}) + l(\mathbf{z}) \\ &\quad s.t.\, \boldsymbol{\mu} - \mathbf{z} = \mathbf{0}_d \end{aligned} \tag{20}$$

with $h(\boldsymbol{\mu}) = (1/2)\|\boldsymbol{y} \odot (\boldsymbol{B}\mu - b_2\mathbf{1}_n) - \mathbf{1}_n\|_2^2$ and $l(\mathbf{z}) = I_{S_0(C_\mu)}(\mathbf{z})$, where $I_{S_0(C_\mu)}(\mathbf{z})$ is the indicator function built on the nonconvex set of C_μ - sparse vectors, and we have $S_0(C_\mu) = \{\mathbf{z} \in \mathbb{R}^d | \|\mathbf{z}\|_0 \le C_\mu\}$.

Since the optimization problem (20) has separable objective and equality constraint functions, it is easy to give the μ - step EM^{2}NOLC algorithm in the scaled ADMM form as below:

$$\boldsymbol{\mu}^{k+1} = \arg\min_{\boldsymbol{\mu}\in\mathbb{R}^d}\left\{h(\boldsymbol{\mu}) + (\rho/2)\left\|\boldsymbol{\mu} - \mathbf{z}^k + \boldsymbol{v}^k\right\|_2^2\right\} \tag{21}$$

$$\mathbf{z}^{k+1} = \arg\min_{\mathbf{z}\in\mathbb{R}^d}\left\{l(\mathbf{z}) + (\rho/2)\left\|\boldsymbol{\mu}^{k+1} - \mathbf{z} + \boldsymbol{v}^k\right\|_2^2\right\} \tag{22}$$

$$\boldsymbol{v}^{k+1} = \boldsymbol{v}^k + \boldsymbol{\mu}^{k+1} - z^{k+1} \tag{23}$$

with the scaled dual variables $\boldsymbol{v}$ ($\boldsymbol{v} \in \mathbb{R}^d$).

Let $\boldsymbol{B}_y = \boldsymbol{y} \odot \boldsymbol{B}$, we define $y \odot \boldsymbol{B} = (y \odot \boldsymbol{B}_1, \cdots, y \odot \boldsymbol{B}_d)$, and $\odot$ is the elementwise product between the label vector $\boldsymbol{y}$ and the column-wise multi-kernel vector $\boldsymbol{B}_m(m = 1, \cdots, d)$. By the KKT optimality condition of the proximal operator (21), the μ - update is written as

$$\boldsymbol{\mu}^{k+1} = \left(\boldsymbol{B}_y^T\boldsymbol{B}_y + \rho\boldsymbol{I}_d\right)^{-1}\left\{\boldsymbol{B}_y^T(b_2\boldsymbol{y} + \mathbf{1}_n) + \rho\left(\mathbf{z}^k - \boldsymbol{v}^k\right)\right\} \tag{24}$$

Similarly, because of $\rho > 0$, the matrix $\boldsymbol{B}_y^T\boldsymbol{B}_y + \rho\boldsymbol{I}_d$ is always invertible.

From the z - minimization (22), the *z−update* can be considered as the orthogonal projection evaluation with the form

$$\mathbf{z}^{k+1} = \Pi_{S_0(C_\mu)}(\boldsymbol{\mu}^{k+1} + \boldsymbol{v}^k) \tag{25}$$

The proof for the derivation of the iterative updates (23), (24), and (25) of the μ - step EM^{2}NOLC algorithm in the scaled ADMM form can be found in "Proof of the μ-step EM^{2}NOLC algorithm via ADMM" in Appendix.

From the v - update (23), the primal residual of the ADMM problem (20) is computed by $\boldsymbol{r}^{k+1}_{\mu-\text{step}} = \boldsymbol{\mu}^{k+1} - \mathbf{z}^{k+1}$. The dual residual can be derived from the augmented Lagrangian function of (20), and it is expressed as $\boldsymbol{s}^{k+1}_{\mu-\text{step}} = \rho(\mathbf{z}^k - \mathbf{z}^{k+1})$. Thus, the stop criteria of the μ - step EM^{2}NOLC algorithm via ADMM are

$$\left\|\boldsymbol{r}^k_{\mu-\text{step}}\right\|_2 \le \varepsilon_{primal} \text{ and } \left\|\boldsymbol{s}^k_{\mu-\text{step}}\right\|_2 \le \varepsilon_{dual} \tag{26}$$

Based on the aforementioned description of the EM^2-NOLC model in ADMM form, the whole EM^{2}NOLC algorithm is summarized into the processing flows in Fig. 1, including inputs, outputs, initialization, the iterative process, decision functions, and value prediction.

4 Experiments

In this section, the ADMM algorithms of LSVMC, SVMC, LASSOC, the MKL algorithms of SimpleMKL and EasyMKL, the GD algorithm of FS-SVMC, and the proposed EM^{2}NOLC are applied to ten benchmark datasets to evaluate their predictive performance by comparison analysis of several statistic measures. The main contents include datasets, experiment design, the comparison of predictive performance, the analysis of instance and feature importance, and the correlation analysis of selected features across folds.

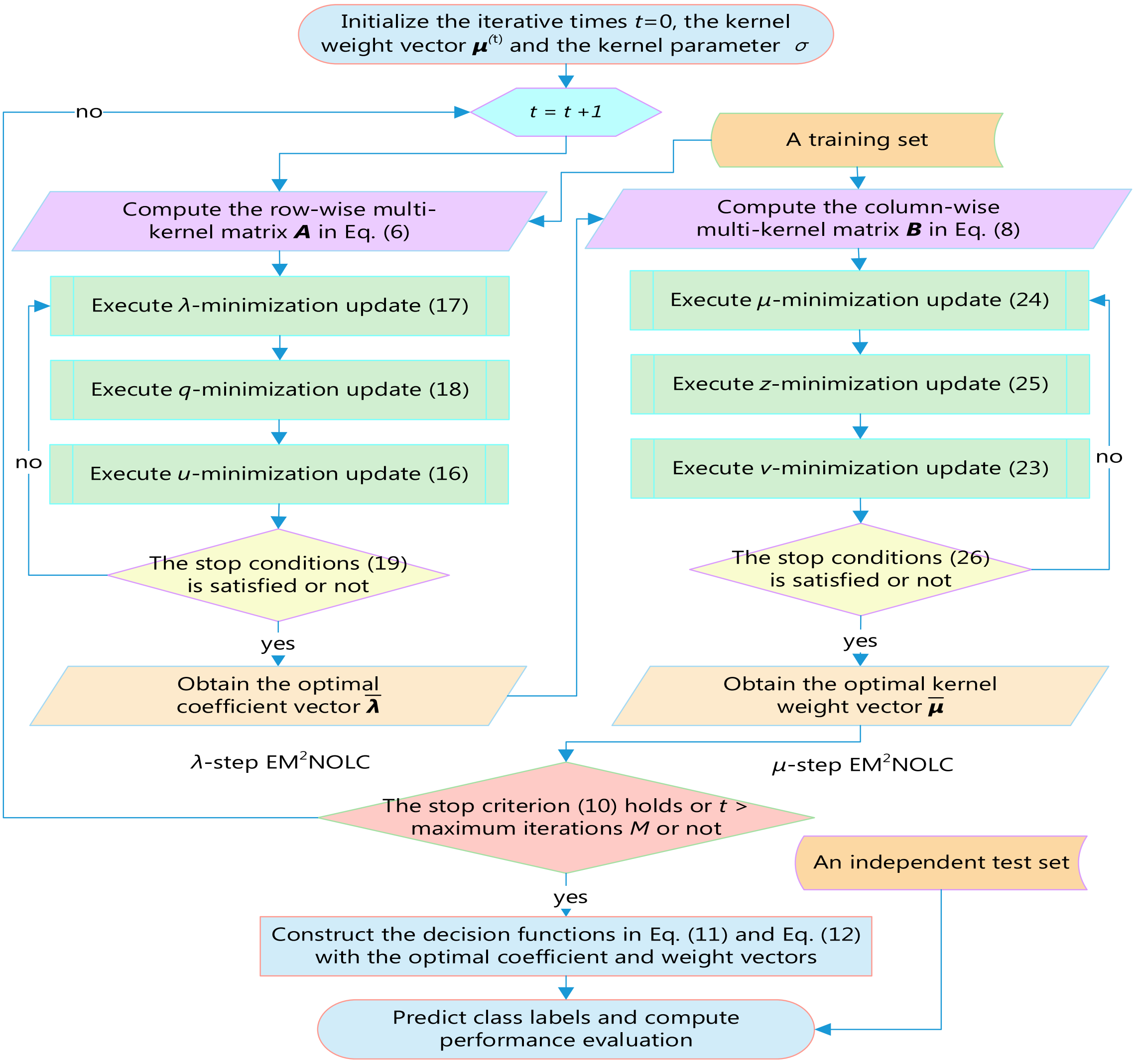

Initialize the iterative times t=0, the kernel weight vector $\boldsymbol{\mu}^{(t)}$ and the kernel parameter σ
no
$t = t + 1$
A training set
Compute the row-wise multi-kernel matrix $\boldsymbol{A}$ in Eq. (6)
Compute the column-wise multi-kernel matrix $\boldsymbol{B}$ in Eq. (8)
Execute λ-minimization update (17)
Execute q-minimization update (18)
Execute u-minimization update (16)
no
The stop conditions (19) is satisfied or not
yes
Obtain the optimal coefficient vector $\bar{\boldsymbol{\lambda}}$
λ-step EM^{2}NOLC
Execute μ-minimization update (24)
Execute z-minimization update (25)
Execute v-minimization update (23)
no
The stop conditions (26) is satisfied or not
yes
Obtain the optimal kernel weight vector $\bar{\boldsymbol{\mu}}$
μ-step EM^{2}NOLC
The stop criterion (10) holds or t > maximum iterations M or not
An independent test set
yes
Construct the decision functions in Eq. (11) and Eq. (12) with the optimal coefficient and weight vectors
Predict class labels and compute performance evaluation

Fig. 1 The processing flows of the EM^{2}NOLC algorithm

4.1 Datasets

In our experiment, ten real datasets are used as benchmarks for evaluating the ADMM algorithms of LSVMC, SVMC, LASSOC, and EM^{2}NOLC, the MKL algorithms of SimpleMKL and EasyMKL, and the GD algorithm of FS-SVMC with predictive accuracy, computational complexity, and interpretability. The datasets used are Diabetes, Heart Disease (HD), German Credit (GC), the early stage of Indian Chronic Kidney Disease (CKD), Wisconsin Diagnostic Breast Cancer (WDBC), Single Proton Emission Computed Tomography image Features of Heart disease (SPECTFH), Parkinson with Replicated Acoustic Features (PRAF), Molecular Splice Junction (MSJ), Musk, Madelon datasets, which are sourced from the online UCI Repository of Machine Learning Databases [52], which is an online database of different datasets with a wide variety of data types. The ten datasets for classification are illustrated in Table 1, including dataset names, the number of instances (#instances), the number of features (#features), and the class-imbalanced ratio (CIR) of the majority to the minority of instances, respectively.

4.2 Experiment design

In this experiment, for the Diabetes, HD, GC, CKD, WDBC, SPECTFH, PRAF, MSJ, Musk, and Madelon datasets, we randomly select 370, 100, 470, 175, 250, 150, 85, 550, 550, and 650 majority-class instances and the same number of minority-class instances, which is in proportion to CIR of the respective dataset, to form the training set by sampling with replacement for the minority class. The remainder is used for the independent test set. Then the fivefold cross-validation (CV) method is used to train LSVMC, SVMC, LASSOC, SimpleMKL, EasyMKL, FS-SVMC, and EM^{2}NOLC on training subsets, the best classifier groups with the optimal parameters are selected on validation subsets, and the averages of predictive performance on the independent test set are calculated and reported. Besides, missing values are filled with the attribute values of the closest neighbour by using the one nearest neighbour (1NN) algorithm. The continuous attributes are discretized by equal width binning with no more than 10 bins to reduce errors in data, and all the attributes are then normalized to the interval [0, 1] using the min–max method.

Table 1 Datasets used for evaluating seven classifiers

Datasets	#Instances	#Features	CIR
Diabetes	768	8	1.87
HD	270	13	1.25
GC	1000	24	2.33
CKD	400	24	1.67
WDBC	567	30	1.68
SPECTFH	267	44	3.85
PRAF	240	45	1.00
MSJ	3190	60	1.10
Musk	6598	166	5.49
Madelon	2600	500	1.00

In the process of training LSVMC, SVMC, LASSOC, SimpleMKL, EasyMKL, FS-SVMC, and EM^{2}NOLC, the grid search method is employed to obtain the best predictive accuracy and the corresponding optimal parameters. That is, the different parametric sets of these classifiers are set in advance, including (i) the regularization parameters C for LSVMC, SVMC, LASSOC, and SimpleMKL from the set $\{\,10^k|k \in \mathbb{Z}\}\,_{k=-1}^{4}$, (ii) the expected numbers of important instances and features C_λ and C_μ for EM^{2}NOLC and the expected numbers of selected features C_m for FS-SVMC from the set $\{\,k|k \in \mathbb{N}\}\,_{k=20}^{1}$, (iii) the bandwidth σ of the RBF kernel from the set $\{\,10^k|k \in \mathbb{Z}\}\,_{k=-2}^{1}$, (iv) the number of weak kernels r for EasyMKL from the set $\{\,r|r = 5k,\, k \in \mathbb{N}\}\,_{k=1}^{20}$, (v) the penalty factor ρ, the smallest constants for τ, ε_{primal}, and ε_{dual}, the maximum iterative times $m(m \in \mathbb{N})$ of EM^{2}NOLC, and the maximum iterative times $t(t \in \mathbb{N})$ of ADMM and GD, are set to 1.0, 1e-4, 50 and 100, respectively, (vi) and the dual variables having the initial vector $\mathbf{0}$ with the same size as the primal variables. For any two input points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ from the training data $\boldsymbol{T}$, the radial basis function (RBF) kernel is used in this study, which is defined as

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 \Big/ 2\sigma^2\right)$$

and the polynomial kernel is applied to SimpleMKL and FS-SVMC, which has the form

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(\boldsymbol{x}_i^T \boldsymbol{x}_j + 1\right)^e$$

with the degree $e \in \{1, 2\}$.

For each of seven classifiers, the number of parametric sets determines the number of nested outer loops, while the size of each parametric set determines the iterative times of each loop. Then the fivefold CV approach is applied in the innermost loop so as to find the optimal parametric combination. The optimal parameters of a classifier group on the training subsets are selected according to the best performance on the validation subsets, so we have five classifiers corresponding five training subsets for the fivefold CV method. Then these classifiers are tested on the independent test set for the prediction of class labels, respectively. The above training and test process of different classifiers are repeated for ten times and the best classifier group with the best predictive performance is selected. Thus, for the performance evaluation and comparative analysis of different classifiers, the averages and standard deviations of predictive accuracies of five classifiers is computed and reported.

In our experiment, several statistical measures are used to evaluate the predictive performance of classifiers, and the Test Accuracy (the total classification accuracy rate on each test set, TA) is defined as

$$TA = \left(1/\left|\cup_{k=1}^{2} I_k\right|\right) \sum_{k=1}^{2} \sum_{i \in I_k} 1(\hat{y}_i = y_i) \times 100\%$$

where $\hat{y}_i$ is the predicted class label of the input point $\boldsymbol{x}_i$ from an independent test set, y_i is the actual class label of the input point $\boldsymbol{x}_i$, $I_1 = \{i | y_i = -1\}$, $I_2 = \{i | y_i = 1\}$, $\cup$ is the union of sets, and $1(\cdot)$ is an indicator function, which takes the value 1 for $\hat{y}_i = y_i$ and 0 otherwise.

The Kolmogorov–Smirnov (KS) statistic is one of the most useful and general nonparametric methods for evaluating classifiers, as it is sensitive to differences in both location and shape of the empirical cumulative distribution functions of two classes. The area under the curve (AUC) statistic is an empirical measure of classification performance based on the area under a receiver operating characteristic (ROC) curve [53], where a classifier is preferred if its ROC curve is closer the upper-left corner, that is, with a large AUC. The value of AUC lies in the interval [0, 1], and a larger AUC means a better predictive performance of a classifier. Besides, we have collected the CPU time in seconds of training (trTime) and test (tsTime) in this experiment.

In terms of the optimal coefficient vector $\overline{\lambda}$ and weight vector $\overline{\mu}$ obtained from the EM^{2}NOLC algorithm, for the instance $\boldsymbol{x}_i$ and the feature $\boldsymbol{f}_m$ in the training set $\boldsymbol{T}$, the instance importance (II) and the feature importance (FI) for classification can be, respectively, computed by

$$\text{II } (\boldsymbol{x}_i) = \left(\overline{\lambda}_i \Big/ \|\overline{\lambda}\|_1\right) \times 100\% \ , \ i = 1, \cdots, n \tag{27}$$

$$\text{FI } (\boldsymbol{f}_m) = \left(\overline{\mu}_m / \|\overline{\mu}\|_1\right) \times 100\% \ , \ m = 1, \cdots, d \tag{28}$$

If the absolute value of II of the input point $\boldsymbol{x}_i$ is not equal to 0, then it is considered as an important instance. The larger the absolute value of II is, the more important the input point $\boldsymbol{x}_i$ is. Otherwise, the input point $\boldsymbol{x}_i$ may be noise or redundancy. Similarity, if the absolute value of FI of the feature $\boldsymbol{f}_m$ is greater than 0, then it is regarded as an important feature. The larger the absolute value of FI is, the more important the feature $\boldsymbol{f}_m$ is. Otherwise, the feature $\boldsymbol{f}_m$ is unimportant and redundant. Those unimportant instances and features can be ignored and removed from the training set $\boldsymbol{T}$. The positive percentage of II or FI shows that the corresponding $\boldsymbol{x}_i$ or $\boldsymbol{f}_m$ has the positive contribution to classification. Otherwise, it has the negative one. Therefore, based on the reduced instance and feature sets, the computational efficiency and interpretability of the EM^{2}NOLC model are improved in the real-world applications.

Finally, all experiments of the LSVMC, SVMC, LASSOC, SimpleMKL, EasyMKL, FS-SVMC, and EM^{2}NOLC methods are implemented on the MATLAB 9.8 platform [54]. That is, the ADMM algorithm of EM^{2}NOLC[1] is programmed by utilizing MATLAB functions we define. The MKL algorithms of SimpleMKL and EasyMKL are, respectively, from the MATLAB toolboxes [9] and [16] and the GD algorithm of FS-SVMC is written as MATLAB functions we define. The ADMM algorithms of LSVMC, SVMC, and LASSOC are sourced from the online MATLAB scripts [36, 55].

4.3 Comparison of predictive accuracies

On ten benchmark datasets, we employ the combining grid search and fivefold CV method to train the ADMM algorithms of LSVMC, SVMC with the RBF kernel, LASSOC, and EM^{2}NOLC with the RBF kernel models, the MKL algorithms of SimpleMKL and EasyMKL, and the GD algorithm of FS-SVMC on training subsets, and validate them on validation subsets in order. Then the classifier groups with the best predictive performance are selected and tested on the independent test sets, respectively. Thus, for TA, KS, AUC of these classifiers, the percentages of their averages and standard deviations are computed and reported in Tables 2, 3, and 4, respectively.

From the TA statistics demonstrated in Table 2, EM^{2}NOLC generally outperforms the other six classifiers of LSVMC, SVMC, LASSOC, SimpleMKL, EasyMKL, and FS-SVMC on the eight independent test sets of Diabetes, HD, GC, WDBC, SPECTFH, PRAF, MSJ, and Madelon, except that it has the same TA statistics as other three classifiers of SimpleMKL, EasyMKL, and FS-SVMC on the CKD test set and it has the second-best TA on the Musk test set.

As shown in Table 3, EM^{2}NOLC generally gives a better KS than that of six classifiers on eight independent test sets. Specifically, the four classifiers of SimpleMKL, EasyMKL, FS-SVMC, and EM^{2}NOLC have the same KS statistics on the CKD dataset. On the Musk dataset, LSVMC has the best KS measure, while EM^{2}NOLC is in the second position and it is followed by the five classifiers of SVMC, LASSOC, EasyMKL, FS-SVMC, and SimpleMKL.

From the AUC statistics shown in Table 4, EM^{2}NOLC generally outperforms other six classifiers on eight independent test sets. Similarly, the four classifiers of

[1] https://github.com/sagedavid/EM2NOLC.

Table 2 The TA (%) comparison of seven classifiers on ten test sets

Classifiers	Datasets									
	Diabetes	HD	GC	CKD	WDBC	SPECTFH	PRAF	MSJ	Musk	Madelon
LSVMC	76.11 ± 0.62	84.44 ± 1.28	68.76 ± 0.92	99.50 ± 0.53	96.92 ± 0.88	76.62 ± 2.33	74.29 ± 3.02	74.81 ± 0.58	**99.71** ± 0.03	99.89 ± 0.05
SVMC	76.16 ± 0.75	83.33 ± 2.03	69.24 ± 0.87	95.16 ± 0.20	96.21 ± 0.41	78.18 ± 1.40	85.43 ± 1.96	78.78 ± 0.43	97.85 ± 0.64	99.94 ± 0.03
LASSOC	74.34 ± 1.03	88.11 ± 0.75	69.64 ± 0.85	97.17 ± 0.99	97.16 ± 0.49	78.70 ± 2.61	71.14 ± 1.96	76.24 ± 0.44	96.46 ± 0.42	99.92 ± 0.00
SimpleMKL	76.87 ± 0.62	84.44 ± 1.05	69.88 ± 0.96	100.0 ± 0.00	92.19 ± 0.61	66.62 ± 2.45	80.67 ± 1.61	74.31 ± 0.42	85.72 ± 0.96	99.72 ± 0.09
EasyMKL	74.75 ± 1.60	84.67 ± 1.64	70.97 ± 2.03	100.0 ± 0.00	91.83 ± 0.87	73.25 ± 2.39	79.00 ± 2.84	66.50 ± 1.12	95.90 ± 0.64	93.09 ± 0.53
FS-SVMC	74.95 ± 0.54	81.89 ± 4.13	68.39 ± 1.08	100.0 ± 0.00	96.86 ± 0.84	61.30 ± 1.71	72.33 ± 2.85	59.45 ± 6.06	94.79 ± 1.72	81.15 ± 4.12
EM^{2}NOLC	**81.52** ± 0.58	**95.56** ± 2.57	**96.73** ± 2.42	100.0 ± 0.00	**99.88** ± 0.26	**91.17** ± 4.81	**98.00** ± 2.39	**84.87** ± 3.31	98.54 ± 0.58	**99.98** ± 0.03

Note that the bold statistics show that the current classifier outperforms others

Table 3 The KS (%) comparison of seven classifiers on ten test sets

Classifiers	Datasets									
	Diabetes	HD	GC	CKD	WDBC	SPECTFH	PRAF	MSJ	Musk	Madelon
LSVMC	49.14 ± 1.27	69.53 ± 2.14	33.17 ± 2.02	98.45 ± 1.10	94.10 ± 1.90	39.80 ± 6.76	51.86 ± 5.99	49.58 ± 1.17	**98.94** ± 0.11	99.78 ± 0.10
SVMC	48.18 ± 1.45	69.80 ± 3.90	36.89 ± 1.18	88.63 ± 0.38	93.04 ± 0.65	38.49 ± 4.41	74.22 ± 3.92	57.75 ± 0.86	90.61 ± 4.48	99.87 ± 0.06
LASSOC	44.59 ± 2.30	75.87 ± 1.45	33.37 ± 1.68	93.56 ± 2.08	94.61 ± 1.18	40.20 ± 6.52	46.55 ± 3.52	52.55 ± 0.90	81.74 ± 1.81	99.85 ± 0.00
SimpleMKL	48.99 ± 1.34	68.46 ± 2.12	35.19 ± 3.04	100.0 ± 0.00	82.83 ± 1.22	22.70 ± 2.97	62.63 ± 3.62	55.63 ± 0.40	50.74 ± 1.85	99.45 ± 0.18
EasyMKL	50.65 ± 2.06	71.43 ± 3.18	37.77 ± 2.89	100.0 ± 0.00	81.59 ± 1.75	34.78 ± 7.49	58.97 ± 5.82	36.58 ± 2.28	79.57 ± 2.68	86.41 ± 1.02
FS-SVMC	45.53 ± 1.19	64.17 ± 8.01	34.07 ± 1.56	100.0 ± 0.00	93.31 ± 1.34	28.77 ± 2.14	56.78 ± 5.23	19.19 ± 12.42	76.27 ± 6.66	62.63 ± 8.29
EM^{2}NOLC	**59.24** ± 1.32	**92.36** ± 4.34	**92.78** ± 4.97	100.0 ± 0.00	**99.68** ± 0.71	**72.18** ± 13.93	**96.08** ± 4.76	**71.79** ± 4.40	94.67 ± 2.58	**99.97** ± 0.07

Table 4 The AUC (%) comparison of seven classifiers on ten test sets

Classifiers	Datasets									
	Diabetes	HD	GC	CKD	WDBC	SPECTFH	PRAF	MSJ	Musk	Madelon
LSVMC	78.43 ± 0.45	83.98 ± 1.55	64.87 ± 0.76	99.53 ± 0.99	96.47 ± 1.16	66.95 ± 5.57	77.25 ± 3.58	75.07 ± 0.47	**98.90** ± 0.10	99.88 ± 0.08
SVMC	72.62 ± 0.35	81.83 ± 2.25	71.48 ± 0.86	92.71 ± 0.24	95.17 ± 0.29	69.33 ± 2.98	83.69 ± 2.22	78.44 ± 0.43	95.30 ± 2.32	99.93 ± 0.04
LASSOC	72.53 ± 0.79	85.70 ± 1.19	66.68 ± 0.10	97.45 ± 1.49	96.58 ± 0.47	69.96 ± 2.33	73.33 ± 2.34	75.59 ± 0.52	90.80 ± 1.02	99.89 ± 0.00
SimpleMKL	72.77 ± 0.42	80.10 ± 1.43	68.19 ± 2.10	100.0 ± 0.00	90.13 ± 1.02	62.04 ± 2.88	82.23 ± 2.03	77.23 ± 0.25	75.56 ± 0.99	99.67 ± 0.12
EasyMKL	72.43 ± 1.71	81.58 ± 2.00	70.08 ± 1.60	100.0 ± 0.00	91.07 ± 1.45	70.01 ± 5.04	77.40 ± 3.26	68.18 ± 1.24	88.97 ± 1.20	92.84 ± 0.55
FS-SVMC	68.93 ± 0.89	80.26 ± 3.36	68.54 ± 1.23	100.0 ± 0.00	96.63 ± 0.67	60.26 ± 2.53	81.24 ± 1.98	60.11 ± 6.16	87.97 ± 3.68	81.68 ± 4.11
EM^{2}NOLC	**81.09** ± 0.99	**96.90** ± 2.19	**96.05** ± 2.19	100.0 ± 0.00	**99.76** ± 0.54	**85.49** ± 8.38	**98.20** ± 2.51	**85.81** ± 2.47	97.37 ± 1.30	**100.0** ± 0.00

SimpleMKL, EasyMKL, FS-SVMC, and EM^{2}NOLC have the same AUC statistics on the CKD dataset. On the Musk dataset, LSVMC is in first place for AUC, and EM^{2}NOLC is in the second place, which is followed by SVMC, LASSOC, EasyMKL, FS-SVMC, and SimpleMKL.

In order to evaluate the runtime performance of seven classifiers, the averages of trTime, and tsTime are collected and reported in Tables 5, and 6, respectively.

From the trTime statistics shown in Table 5, LASSOC takes the least amount of CPU time to train the corresponding classifier model on the ten training sets, while LSVMC spends the most CPU time to learn its classifier models on the nine training sets except for the Madelon datasets. In general, FS-SVMC has the second-most trTime on the nine training sets other than on the Madelon training set, and SimpleMKL is the third-most trTime on the six training sets. The proposed EM^{2}NOLC is in the middle for the trTime statistic.

As shown in Table 6, LSVMC spends the least amount of tsTime on the five test sets of Diabetes, GC, WDBC, SPECTFH, and Madelon. LASSOC has the least amount of tsTime on the MSJ and Musk test sets and EasyMKL has the minimal tsTime on the HD and PRAF test sets. SimpleMKL takes the minimum amount of tsTime on the CKD test set. However, the proposed EM^{2}NOLC has the slightly more tsTime than other six classifiers owing to the computational cost of multi-kernel matrix in the decision function.

4.4 Visual analysis of predictive performance

To obtain the intuitive and vivid comparison of the predictive performance of four ADMM classifiers, the classifier with the best predictive performance on each of four medical independent test sets is chosen from the selected group by combining grid search and fivefold CV method. Figure 2 in "KS curves in Fig. 2" in Appendix plots the KS curves of the cumulative distributions of negative patients (NP) and abnormal (positive) patients (AP) predicted by four ADMM classifiers on four medical test sets, which are also called the corresponding the false positive and true positive rates.

As shown in Fig. 2, the KS curves of EM^{2}NOLC achieves the maximum differences between the cumulative distributions of two classes on four medical test sets. We can see that the outermost KS curves almost belonged to EM^{2}NOLC on four independent test sets. From the visual description of KS curves, LSVMC is in the second place followed by LASSOC and SVMC on the CKD test set. On the PRAF test set, SVMC obtains the best KS statistic except for EM^{2}NOLC, but LASSOC gives the worst KS curves, where the true positive and false positive rate curves of the cumulative distributions of two classes are

Table 5 The trTime (seconds) comparison of seven classifiers on ten training sets

Classifiers	Datasets									
	Diabetes	HD	GC	CKD	WDBC	SPECTFH	PRAF	MSJ	Musk	Madelon
LSVMC	413.7899	679.5037	353.389	297.8363	288.5581	327.5436	440.1959	419.2341	870.7205	537.6537
SVMC	20.1083	0.9539	12.1379	0.2717	4.0845	1.5633	0.0771	18.4043	16.8405	25.9512
LASSOC	**0.0066**	**0.0119**	**0.0128**	**0.0076**	**0.0067**	**0.0068**	**0.0092**	**0.0138**	**0.1248**	**0.1044**
SimpleMKL	4.7509	0.7997	19.2575	2.6400	4.4869	1.7466	1.4449	4.2741	12.6966	2403.6860
EasyMKL	0.2241	0.0635	0.5066	0.1322	0.1990	0.0564	0.0232	0.8221	0.9804	1.3031
FS-SVMC	41.3417	14.3046	99.9939	7.6868	39.9789	16.1496	13.6747	305.9474	607.3566	1265.6150
EM^{2}NOLC	4.8888	0.1446	6.2517	1.0730	5.0537	1.6400	0.7889	57.9348	153.1843	31.9867

Note that the bold statistics show that the current classifier takes less CPU time than others

Table 6 The tsTime (seconds) comparison of seven classifiers on ten test sets

Classifiers	Datasets									
	Diabetes	HD	GC	CKD	WDBC	SPECTFH	PRAF	MSJ	Musk	Madelon
LSVMC	**0.0002**	0.0048	**0.0002**	0.0042	**0.0002**	**0.0002**	0.0041	0.0055	0.0122	**0.0081**
SVMC	0.0279	0.0050	0.0489	0.0060	0.0044	0.0032	0.0023	0.3759	3.1029	0.3461
LASSOC	0.0005	0.0055	0.0003	0.0031	0.0033	0.0033	0.0045	**0.0037**	**0.0115**	0.0106
SimpleMKL	0.0082	0.0025	0.0228	**0.0013**	0.0019	0.0027	0.0023	1.1833	8.5818	0.5210
EasyMKL	0.0302	**0.0022**	0.0248	0.0036	0.0055	0.0020	**0.0013**	0.2292	0.5750	0.1534
FS-SVMC	0.0049	0.0009	0.0026	0.0158	0.0049	0.0008	0.0051	0.0249	0.0733	0.0459
EM^{2}NOLC	0.0522	0.0080	0.0574	0.0152	0.0524	0.0216	0.0111	3.4012	29.2507	23.8968

partially crossed in the lower left part. On the SPECTFH test set, the KS curves of LASSOC generates the second-largest distance, but SVMC generates the minimal one. Except for EM^{2}NOLC, the KS curves of the other three classifiers provide the similar differences, which means that they have almost similar KS statistics on the WDBC test set.

Based on the selected classifiers with the best predictive performance, Fig. 3 in "ROC curves in Fig. 3" in Appendix draws the ROC curves of four ADMM classifiers on the four independent medical test sets.

As demonstrated in Fig. 3, we find that the ROC curves of EM^{2}NOLC enclose the maximal areas on four medical test sets. Specifically, LSVMC generates the second-largest AUC followed by LASSOC, and SVMC generates the smallest AUC on the CKD test set. For the areas under the ROC curves SVMC is in the second place followed by LSVMC, and LASSOC is in last on the PRAF test set. On the SPECTFH test set, the ROC curve of LASSOC gives the second-largest area, but LSVMC ranks last, where SVMC is in the middle of LASSOC and LSVMC. Similar to the SPECTFH test set, the same rank list is coincidentally obtained on the WDBC test set. That is, LASSOC gives the second-largest AUC, and then SVMC is in the middle of LASSOC and LSVMC, LSVMC provides the smallest AUC on the independent test set.

4.5 Quantity comparison of important instances and features

Since the number of the important instances (#IIs) and the number of the important features (#IFs) are set in advance, EM^{2}NOLC automatically extracts the specified quantity of important instances (IIs) and features (IFs). In this experiment, we have $\#IIs = C_{\lambda}$ and $\#IFs = C_{\mu}$ for EM^{2}NOLC. For SVC, SimpleMKL, EasyMKL, and FS-SVMC, important instances are called support vectors (SVs), so we have #IIs = #SVs, where their Lagrange multipliers are greater than zero and less than the regularization parameter C. To evaluate the ability of classifiers to identify the important instances and features from the training set, we have to define the instance reduction rate (IRR) and the feature reduction rate (FRR) with respect to #IIs and #IFs. That is, the ratios of the number of unimportant instances to the size of training set and the proportion of the number of unimportant features to the size of the feature set, are expressed as

$$\mathrm{IRR} = 1 - (\#\mathrm{IIs}/|\boldsymbol{X}|) \times 100\%$$

$$\mathrm{FRR} = 1 - (\#\mathrm{IFs}/|\boldsymbol{F}|) \times 100\%$$

Based on the selected classifier group by using the grid search and fivefold CV method, the averages of #IIs (SVs) and #IFs and the percentage statistics of IRR and FRR on ten training sets are reported in Tables 7 and 8, respectively.

As shown in Table 7, EM^{2}NOLC generally selects the least number of IIs and IFs from ten training sets, compared with the other six classifiers. Specifically, LASSOC extracts the minimum number of IFs from the Diabetes training set and FS-SVMC identifies the minimum number of IFs from the HD and SPECTFH, respectively. SVM, SimpleMKL, EasyMKL, and FS-SVMC can extract IIs from ten training sets, but the number of IIs given by them is far more than that of our proposed ADMM classifier. LASSOC and FS-SVMC can select IFs from ten training sets, but the number of IFs selected by them is more than that of the EM^{2}NOLC algorithm on seven training sets, except for the Diabetes, HD, and SPECTFH training sets.

As shown in Table 8, EM^{2}NOLC generally provides the highest reduction rates of instances (IRR) and features (FRR) on ten training sets. Similar to #IFs, LASSOC obtains the largest FRR on the Diabetes training set, and FS-SVMC has the largest FRR on the HD and SPECTFH training sets, respectively. Since SVMC, SimpleMKL, EasyMKL, and FS-SVMC can extract IIs, they have the nonzero IRR, unless they use the whole training set with the zero IRR. However, they are unable to select IFs, where FRR is zero. Since LASSOC and FS-SVMC can IFs, they have nonzero FRR, only if they employ the whole training set with the zero FRR. But they have no ability to identify IIs, where IRR is zero. LSVMC has no capability to extract either IIs or IFs.

4.6 Analysis of instance and feature importance

Here we take the four medical datasets of CKD, PRAF, SPECTFH, and WDBC for example, according to the selected EM^{2}NOLC with the best predictive performance on each of four test sets, the optimal coefficient vector $\overline{\lambda}$ and kernel weight vector $\overline{\mu}$ are obtained on each of four training sets. Then we can obtain the important instances

Table 7 Comparison of #IIs and #IFs for seven classifiers on ten training sets

Classifiers	Datasets																			
	Diabetes		HD		GC		CKD		WDBC		SPECTFH		PRAF		MSJ		Musk		Madelon	
	#IIs	#IFs	#IIs	#IFs	#IIs	#IFs	#IIs	#IFs	#IIs	#IFs	#IIs	#IFs	#IIs	#IFs	#IIs	#IFs	#IIs	#IFs	#IIs	#IFs
LSVMC	592	8	160	13	752	24	350	24	500	30	300	44	170	45	880	60	880	166	1040	500
SVMC	405	8	114.9	13	746.6	24	280	24	112.2	30	172.2	44	128.8	45	734.7	60	576.6	166	285.8	500
LASSOC	592	**3.6**	160	10	752	13.2	350	15.4	500	21.8	300	28.8	170	24.8	880	54.9	880	70.3	1040	278.6
SimpleMKL	545.4	8	137.8	13	672.4	24	158.8	24	213.9	30	237.8	44	134.9	45	782.4	60	770.4	166	261	500
EasyMKL	520.4	8	157.6	13	735	24	170.4	24	297.4	30	224.8	44	134.2	45	880	60	684	166	1039	500
FS-SVMC	592	8	160	**7**	752	13	37.8	20	61.8	20	270	**6**	162	20	880	20	157.4	20	1040	20
EM2NOLC	**17**	5	**9**	10	**20**	**10**	**1**	**4**	**17**	**2**	**8**	19	**11**	**1**	**15**	**4**	**3**	**16**	**14**	**14**

Note that the bold statistics show that the current classifier selects the least number of #IIs and #IFs than the other seven classifiers

Table 8 Comparison of IRR (%) and FRR (%) for seven classifiers on ten training sets

Classifiers	Datasets									
	Diabetes		HD		GC		CKD		WDBC	
	IRR	FRR	IRR	FRR	IRR	FRR	IRR	FRR	IRR	FRR
LSVMC	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SVMC	31.59	0.00	28.38	0.00	0.72	0.00	20.00	0.00	77.56	0.00
LASSOC	0.00	**55.00**	0.00	23.08	0.00	45.00	0.00	35.83	0.00	27.33
SimpleMKL	7.87	0.00	13.88	0.00	10.59	0.00	54.63	0.00	57.22	0.00
EasyMKL	12.09	0.00	1.50	0.00	2.26	0.00	51.31	0.00	40.52	0.00
FS-SVMC	0.00	0.00	0.00	**46.15**	0.00	45.83	89.20	16.67	87.64	33.33
EM2NOLC	**97.13**	37.50	**94.38**	23.08	**97.34**	**58.33**	**99.71**	**83.33**	**96.60**	**93.33**

Classifiers	Datasets									
	SPECTFH		PRAF		MSJ		Musk		Madelon	
	IRR	FRR	IRR	FRR	IRR	FRR	IRR	FRR	IRR	FRR
LSVMC	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SVMC	42.60	0.00	24.24	0.00	16.51	0.00	34.48	0.00	72.52	0.00
LASSOC	0.00	34.55	0.00	44.89	0.00	8.50	0.00	57.65	0.00	44.28
SimpleMKL	20.73	0.00	50.14	0.00	11.09	0.00	12.45	0.00	74.90	0.00
EasyMKL	25.07	0.00	21.06	0.00	0.00	0.00	22.27	0.00	0.10	0.00
FS-SVMC	10.00	**86.36**	4.71	55.56	0.00	66.67	82.11	87.95	0.00	96.00
EM2NOLC	**97.33**	56.82	**93.53**	**97.78**	**98.30**	**93.33**	**99.66**	**90.36**	**98.65**	**97.20**

Note that the bold statistics show that the current classifier obtains the maximal percentage of IRR and FRR than the other seven classifiers

(IIs) and features (IFs) and compute their II and FI values by Eqs. (27) and (28). For each of IIs or IFs, its percentage of II or FI can be regarded as a contribution to classification, and its plus or minus sign means that there is a positive or negative impact on prediction. Thus, we plot the IIs and IFs with their percentage values of II and FI sorted by absolute value in Figs. 4 and 5, which are given in Appendices "IFs for EM2NOLC in Fig. 4" and "IFs for EM2NOLC in Fig. 5", respectively.

As demonstrated in Fig. 4, EM^{2}NOLC extracts a small number of IIs from four medical training sets. Specifically, the only important instance $\boldsymbol{x}_{67}$ gives a significantly positive contribution of 100% to the CKD prediction, which can be regarded as a reference point or prototype. The eleven IIs are identified from the PRAF training set. Two and nine IIs, respectively, have the positive and negative impacts on the prognosis of PRAF. The most important instance is $\boldsymbol{x}_3$(CONT-01–1) has a negative impact of -12.41% on the prediction of PRAF. The eight IIs are selected from the SPECTFH training set. The positive and negative impacts on the SPECTFH prediction come from three and five IIs, respectively. The most important instance $\boldsymbol{x}_4$ has a negative contribution of –21.61% to the prognosis of SPECTFH. The seventeen IIs are extracted from the WDBC training set. The positive and negative impacts on the WDBC prediction are from two and fifteen IIs, respectively. The most important instances $\boldsymbol{x}_{293}$(No. 891670) has a negative contribution of –11.53% to the WDBC prognosis.

As shown in Fig. 5, EM^{2}NOLC selects a small quantity of IFs from four medical training sets. Specifically, there are four IFs identified from the CKD training set. The positive and negative contributions to the CKD prediction come from two and two IFs, respectively. The most important feature $\boldsymbol{f}_4$(albumin) has the negative contribution of –44.42% to the CKD prognosis. The only important feature $\boldsymbol{f}_{42}$(Delta6) gives a 100% positive impact on the classification of PRAF. The nineteen IFs are selected from the SPECTFH training set. The positive and negative contributions to the SPECTFH prediction are from eleven and eight IFs, respectively. The most important feature $\boldsymbol{f}_{42}$(F21S) has a positive contribution of 10.06% to the prognosis of SPECTFH. There are two IFs selected from the WDBC training set. The two IFs have negative contributions to the WDBC prediction and the biggest contribution of –97.43% comes from the most important feature f_8(mean radius of concave points).

Once we compute the weighted similarities between unseen input points and the important instances or features with respect to the sparse coefficients $\overline{\lambda}$ and the sparse weights $\overline{\mu}$, the decision functions in Eq. (11) or Eq. (12) can be used to predict the class labels of unknown input points. Since the contribution to classification for each important instance or feature is quantifiable and traceable, the disease prediction is transparent and explainable in the light of the percentages of impacts or contributions of important instances and features.

4.7 Correlation analysis of selected features across folds

By using the grid search and fivefold CV method, the EM^{2}NOLC classifier group with the best averages of predictive accuracies are determined. The important features in each of the fivefolds are obtained based on the optimal kernel weight vector $\overline{\mu}$ on each of training subsets. Sometimes the import features across fivefolds are all the same, but they usually have both common features and different features. Here taking the GC and MSJ datasets for example, where ten and four important features are, respectively, selected from the GC and MSJ training set (see Table 7), we compute the absolute value of the Pearson correlation coefficient (CC) between any two features across folds. The heatmaps of correlation matrices of selected features across folds are plotted in Figs. 6 and 7, which are given in Appendices "Correlation analysis of selected features in Fig. 6" and " Correlation analysis of selected features in Fig. 7", respectively.

As shown in Fig. 6, the common feature is the feature f_2 across fivefolds, where CC of itself takes the value 1. Specifically, for example, for fold 1 and fold 2, their intersection includes the features of f_2,f_5,f_7,f_9,f_{21}, and f_{22} with CC of 1. The CC takes value of 0.62 between f_4 in fold 1 and f_2 in fold 2, and the CC has the value of 0.74 between f_{20} in fold 1 and f_{21} in fold 2. There is weak correlation or uncorrelation among other features from fold 1 and fold 2, respectively. For fold 4 and fold 5, their common features has features of f_2,f_7, and f_{22} with CC of 1. The CC takes value of 0.62 between f_4 in fold 4 and f_2 in fold 5, while other features across fold 4 and fold 5 have weak correlation or uncorrelation. Similarly, as shown in Fig. 7, the common feature is the feature f_{29} and f_{30} across fivefolds, where CC of themselves takes the value 1. Except the two features f_{29} and f_{30}, any two features across folds have weak or zero correlation.

In addition to the GC and MSJ datasets, experiments on the other eight datasets give similar results. Thus, we find that among selected important features across folds, the common features are composed of the core of the classifier model, and most of other features have weak or zero correlations between them.

5 Discussion

5.1 Competence comparison of classifiers

From the experimental results and comparative analysis of four ADMM, two MKL, one GD classifiers in Sects. 4.3, 4.4, and 4.5, we know that extracting important instances and features not only increases classification performance but also produces traceable and interpretable predictions (see Tables 2, 3, 4, 5, 6, 7, 8, and Figs. 2, 3), which are very important to some industries like medical and finance in practice. The analysis of important instances and features and the correlation analysis of selected features in Sects. 4.6 and 4.7 further confirm the before assertion (see Figs. 4, 5, 6, and 7). Specifically, from four perspectives of prediction (yes/no), extraction of IIs (yes/no), and selection of IFs (yes/no), and interpretability (no/partial/complete), the competence comparison of different classifiers is summarized into a Table 9.

As shown in Table 9, all four ADMM classifiers are capable of predicting class labels for given inputs. LSVMC is unable to extract important instances and features. It is noted that LSVMC can obtain weights of different features, but it has no ability to produce sparse weight vectors. Therefore, LSVMC is unexplainable, especially for high-dimensional data. SVMC, SimpleMKL, and EasyMKL can

Table 9 Competence comparison of seven classifiers

Classifiers	Prediction	Extraction of IIs	Selection of IFs	Interpretability
LSVMC	Yes	No	No	No
SVMC	Yes	Yes	No	Partial
LASSOC	Yes	No	Yes	Partial
SimpleMKL	Yes	Yes	No	Partial
EasyMKL	Yes	Yes	No	Partial
FS-SVMC	Yes	Yes	Yes	Complete
EM^{2}NOLC	Yes	Yes	Yes	Complete

extract IIs, so it is partially interpretable. LASSOC can select IFs, thus it is partially interpretable. EM^{2}NOLC and FS-SVMC not only select IIs and IFs but compute their percentage contribution to prediction, hence they are completely interpretable. However, compared with EM^2-NOLC, FS-SVMC has a limited reduction of instances (see Tables 7 and 8). That is, the number of selected instances by FS-SVMC is far more than that of EM^{2}NOLC, so it has weak interpretability for the contribution of important instances to classification.

Besides, for any input point, we need to, respectively, compute row- and column-wise kernel similarities between input point and important instances with important features from the training set. Then weighed similarities with respect to sparse coefficient and kernel weight vectors are obtained, and class labels are predicted according to signs of weighed similarities (see decision functions in Eqs. (11) and (12)). So, the main factors of prediction are completely traceable.

5.2 Analysis of computational complexities

For the time complexity of four ADMM classifiers, the number of their basic operations to solve a problem often is often used to measure the time complexity of algorithms. For LSVMC, the w - update is a critical step, and it employs gradient descent or Newton's method to find optimal solution with the time complexity $O(1/\varepsilon)$ or $O(tdn^3)$, where ε is precision and t ($t \in \mathbb{N}$) is the maximum iteration times of ADMM, d is the dimensional size, and n is the sample size. The α - update of SVMC and the w - update of LASSOC are two main steps, they need to use Cholesky factorization to solve linear equation systems with the time complexity $O(tn^3)$. For EM^{2}NOLC, the λ - update(23) and μ - update(30) are two core steps, but they need to utilize Cholesky decomposition to solve linear equation systems in a serial fashion, with the time complexity $O(mtn^3)$, where m ($m \in \mathbb{N}$) is the maximum iteration times of EM^{2}NOLC.

For the computational complexity of the two MKL classifiers, SimpleMKL combines the MKL technique and quadratic programming algorithm with the active constraint method to solve classification problems with the time complexity $O(kdn^3)$, where $k(k \in \mathbb{N})$ is the maximum iteration times of SimpleMKL. EasyMKL uses the MKL method with weak kernels and quadratic programming algorithm to solve classification problems with the time complexity $O(rn^3)$, where r is the number of weak kernels. For the GD algorithm of FS-SVMC, it employs GD to solve two quadratic programming problems with the time complexity $O(tdn^3)$, where $t(t \in \mathbb{N})$ is the maximum iteration times of GD.

6 Conclusions

In this paper, we propose an EM^{2}NOLC method based on row- and column-wise multi-kernel matrices, and the corresponding algorithm is implemented in the framework of ADMM. The proposed classifier can simultaneously extract important instances and features apart from prediction, and it can provide interpretable and trackable classification. On ten real datasets, EM^{2}NOLC generally achieves better predictive performance and interpretable results than LSVMC, SVMC, LASSOC, SimpleMKL, EasyMKL, and FS-SVMC. Experimental results, the comparison of predictive performance, and the analysis of important instances and features show that EM^{2}NOLC is an effective classification approach. Due to its characteristic of iterative update, EM^{2}NOLC obviously has great potential as a prospective classification approach for other real-world applications. Finally, we plan to use EM^{2}NOLC to address the large-scale and high-dimensional medical diagnosis and prognosis problem, aiming for simultaneous classification and selection of instances and features with high predictive accuracy, efficiency, and interpretability in the next step.

Appendix

Proof of equality (6)

For any two input points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ $(i,j = 1,\cdots,n)$ from the training set $\boldsymbol{T}$, suppose that a basis function $\phi(\cdot)$ mapping any feature value $\boldsymbol{x}_{jm}(m = 1,\cdots,d)$ from the input space to a new feature space is given. If their product $\phi(\boldsymbol{x}_{jm}) \times \phi(\boldsymbol{x}_{im})$ in the feature space can be replaced with the kernel function $\kappa(x_{jm}, x_{im})$ regarding the mth feature. The kernel vector $\boldsymbol{u}_{ji}(\boldsymbol{u}_{ji} \in \mathbb{R}^d)$ of d features is denoted as

$$\begin{aligned}\boldsymbol{u}_{ji} &= \left[\phi(x_{j1}),\cdots,\phi(x_{jd})\right] \odot \left[\phi(x_{i1}),\cdots,\phi(x_{id})\right]\\ &= \left[\phi(x_{j1})\phi(x_{i1}),\cdots,\phi(x_{jd})\phi(x_{id})\right]\\ &= \left[\kappa(x_{j1},x_{i1}),\cdots,\kappa(x_{jd},x_{id})\right].\end{aligned}$$

Given the kernel weight vector $\boldsymbol{\mu}^t$ at iteration t, the row-wise multi-kernel vector $\boldsymbol{A}_j(\boldsymbol{A}_j \in \mathbb{R}^n)$ is defined as a weighted similarity between the input points $\boldsymbol{x}_j$ and other input points in the training set, that is we have the equality

$$\boldsymbol{A}_j = \left(\boldsymbol{u}_{j1}\boldsymbol{\mu}^t, \cdots, \boldsymbol{u}_{jn}\boldsymbol{\mu}^t\right)^T, \ \ j = 1, \cdots, n$$

The row-wise multi-kernel matrix $\boldsymbol{A}(\boldsymbol{A} \in \mathbb{R}^{n\times n})$ has the below form

$$\boldsymbol{A} = (\boldsymbol{A}_1, \cdots, \boldsymbol{A}_n)$$

So, for any two input points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, the element of the matrix $\boldsymbol{A}$ is $\boldsymbol{A}_{ji} = \sum_{m=1}^{d} \mu_m^t \kappa(\boldsymbol{x}_{jm}, \boldsymbol{x}_{im})$ for all $i,j = 1, \cdots, n$.

Proof of equality (8)

For any feature $\boldsymbol{f}_m(\boldsymbol{f}_m \in \mathbb{R}^n$, $m = 1, \cdots, d)$ from the training set $\boldsymbol{T}$, assume that the mapping function $\phi(\cdot)$ transforms the feature value $\boldsymbol{x}_{jm}(j = 1, \cdots, n)$ in the input space into a new feature space, the Kronecker product $\boldsymbol{P}$ $(\boldsymbol{P} \in \mathbb{R}^{n\times n})$ regarding feature $\boldsymbol{f}_m$ is computed by

$$\begin{aligned}\boldsymbol{P} &= \boldsymbol{f}_m \boldsymbol{f}_m^T \\ &= [\phi(x_{1m}), \cdots, \phi(x_{nm})]^T [\phi(x_{1m}), \cdots, \phi(x_{nm})] \\ &= \begin{pmatrix} \phi(x_{1m})\phi(x_{1m}) & \ldots & \phi(x_{1m})\phi(x_{nm}) \\ \vdots & \ddots & \vdots \\ \phi(x_{nm})\phi(x_{1m}) & \cdots & \phi(x_{nm})\phi(x_{nm}) \end{pmatrix} \\ &= \begin{pmatrix} \kappa(x_{1m}, x_{1m}) & \ldots & \kappa(x_{1m}, x_{nm}) \\ \vdots & \ddots & \vdots \\ \kappa(x_{nm}, x_{1m}) & \cdots & \kappa(x_{nm}, x_{nm}) \end{pmatrix}.\end{aligned}$$

If the coefficient vector $\boldsymbol{\lambda}^t$ at iteration t is given, then the column-wise multi-kernel vector $\boldsymbol{B}_m(\boldsymbol{B}_m \in \mathbb{R}^n)$ with respect to the mth feature can be obtained by the multiplication of the transposed matrix $\boldsymbol{P}^T$ and the vector $\boldsymbol{\lambda}^t$ with the form

$$\begin{aligned}\boldsymbol{B}_m &= \boldsymbol{P}^T \boldsymbol{\lambda}^t \\ &= \begin{pmatrix} \kappa(x_{1m}, x_{1m}) & \ldots & \kappa(x_{1m}, x_{nm}) \\ \vdots & \ddots & \vdots \\ \kappa(x_{nm}, x_{1m}) & \cdots & \kappa(x_{nm}, x_{nm}) \end{pmatrix}^T \begin{bmatrix} \lambda_1^t \\ \vdots \\ \lambda_n^t \end{bmatrix} \\ &= \left[\sum_{j=1}^{n} \lambda_j^t \kappa(x_{jm}, x_{1m}), \cdots, \sum_{j=1}^{n} \lambda_j^t \kappa(x_{jm}, x_{nm})\right]^T.\end{aligned}$$

The column-wise multi-kernel matrix $\boldsymbol{B}(\boldsymbol{B} \in \mathbb{R}^{n\times d})$ is denoted as.

$$\boldsymbol{B} = (\boldsymbol{B}_1, \cdots, \boldsymbol{B}_d).$$

Thus, for any input points $\boldsymbol{x}_i$ with respect to the feature $\boldsymbol{f}_m$, the element of the matrix $\boldsymbol{B}$ is $\boldsymbol{B}_{im} = \sum_{j=1}^{n} \lambda_j^t \kappa(\boldsymbol{x}_{jm}, \boldsymbol{x}_{im})$ for all $i = 1, \cdots, n$ and $m = 1, \cdots, d$.

Proof of the $\boldsymbol{\lambda}-$step EM^{2}NOLC algorithm via ADMM

Corresponding with the λ - step EM^{2}NOLC model (7) and its ADMM optimization problem (13) with the separable objective and equality constraint functions, the augmented Lagrangian function regarding the scaled dual variables $\boldsymbol{u}(\boldsymbol{u} \in \mathbb{R}^n)$ and the penalty parameter $\rho(\rho > 0)$ is defined as.

$$\boldsymbol{L}_\rho(\boldsymbol{\lambda}, \boldsymbol{q}, \boldsymbol{u}) = f(\boldsymbol{\lambda}) + g(\boldsymbol{q}) + (\rho/2)\|\boldsymbol{\lambda} - \boldsymbol{q} + \boldsymbol{u}\|_2^2 - (\rho/2)\|\boldsymbol{u}\|_2^2$$

The ADMM updates (17), (18), and (16) of the λ - step EM^{2}NOLC algorithm are obtained from the partial derivative of $\boldsymbol{L}_\rho(\lambda, \boldsymbol{q}, \boldsymbol{u})$ with respect to its three parameters $\boldsymbol{\lambda}$, $\boldsymbol{q}$, and $\boldsymbol{u}$, respectively. We can express the λ - minimization as the proximal operator:

$$\begin{aligned}\boldsymbol{\lambda}^{k+1} &= \underset{\lambda \in \mathbb{R}^n}{\arg\min}\, \boldsymbol{L}_\rho(\boldsymbol{\lambda}, \boldsymbol{q}^k, \boldsymbol{u}^k) \\ &= \underset{\lambda \in \mathbb{R}^n}{\arg\min} \left\{ f(\boldsymbol{\lambda}) + (\rho/2)\left\|\boldsymbol{\lambda} - \boldsymbol{q}^k + \boldsymbol{u}^k\right\|_2^2 \right\}.\end{aligned}$$

Then the gradient of $\boldsymbol{L}_\rho(\boldsymbol{\lambda}, \boldsymbol{q}^k, \boldsymbol{u}^k)$ regarding $\boldsymbol{\lambda}$ is set to zero, we analytically obtained the resulting λ - update:

$$\begin{aligned}&\nabla_\lambda \boldsymbol{L}_\rho(\boldsymbol{\lambda}, \boldsymbol{q}^k, \boldsymbol{u}^k) = 0 \\ &\Rightarrow \nabla_\lambda \left\{ f(\boldsymbol{\lambda}) + (\rho/2)\left\|\boldsymbol{\lambda} - \boldsymbol{q}^k + \boldsymbol{u}^k\right\|_2^2 \right\} = 0 \\ &\Rightarrow \nabla_\lambda \left\{ (1/2)\|\boldsymbol{y} \odot (\boldsymbol{A}\lambda - b_1 \mathbf{1}_n) - 1_n\|_2^2 + (\rho/2)\left\|\boldsymbol{\lambda} - \boldsymbol{q}^k + \boldsymbol{u}^k\right\|_2^2 \right\} = 0 \\ &\Rightarrow \boldsymbol{A}_y^T \boldsymbol{A}_y \boldsymbol{\lambda} - \boldsymbol{A}_y^T (b_1 \boldsymbol{y} + \mathbf{1}_n) + \rho\lambda - \rho\left(\boldsymbol{q}^k - \boldsymbol{u}^k\right) = 0 \\ &\Rightarrow \lambda^{k+1} = \left(\boldsymbol{A}_y^T \boldsymbol{A}_y + \rho I_n\right)^{-1} \left\{ \boldsymbol{A}_y^T (b_1 y + \mathbf{1}_n) + \rho\left(\boldsymbol{q}^k - \boldsymbol{u}^k\right) \right\}.\end{aligned}$$

The q - minimization of the λ - step EM^{2}NOLC algorithm has the form

$$\begin{aligned}\boldsymbol{q}^{k+1} &= \underset{\boldsymbol{q} \in \mathbb{R}^n}{\arg\min}\, \boldsymbol{L}_\rho(\lambda^{k+1}, \boldsymbol{q}, \boldsymbol{u}^k) \\ &= \underset{\boldsymbol{q} \in \mathbb{R}^n}{\arg\min} \left\{ g(\boldsymbol{q}) + (\rho/2)\left\|\lambda^{k+1} - \boldsymbol{q} + \boldsymbol{u}^k\right\|_2^2 \right\}.\end{aligned}$$

Dual to $S_0(C_\lambda) = \{\boldsymbol{q} \in \mathbb{R}^n | \|\boldsymbol{q}\|_0 \leq C_\lambda\}$ is a nonconvex set, the q - minimization may not converge to an optimal point. We can apply the projected gradient method to approximate the q - update procedure. So, we have the q - update:

$$
\begin{aligned}
\boldsymbol{q}^{k+1} &= \mathop{\arg\min}_{\boldsymbol{q}\in\mathbb{R}^n}\left\{I_{S_0(C_\lambda)}(\boldsymbol{q}) + (\rho/2)\left\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{q} + \boldsymbol{u}^k\right\|_2^2\right\}\\
&\approx \mathop{\arg\min}_{\boldsymbol{q}\in S_0(C_\lambda)}\left\|\boldsymbol{\lambda}^{k+1} + \boldsymbol{u}^k - \boldsymbol{q}\right\|_2^2\\
&= \Pi_{S_0(C_\lambda)}(\boldsymbol{\lambda}^{k+1} + \boldsymbol{u}^k),
\end{aligned}
$$

where the indicator function has $I_{S_0(C_\lambda)}(\boldsymbol{q}) = 1$ if $\boldsymbol{q} \in S_0(C_\lambda)$, otherwise $I_{S_0(C_\lambda)}(\boldsymbol{q}) = 0$. For any vector $\boldsymbol{q}(\boldsymbol{q} \in \mathbb{R}^n)$, the projection operator $\Pi_{S_0(C_\lambda)}(\boldsymbol{q})$ can be actually implemented by sorting the elements of the vector $\boldsymbol{q}$ in descending order of their absolute values and setting all elements to zeros except top C_λ.

The u - update step can be considered as the change of constraint residuals in the optimization problem (13), which ensures the convergence of the ADMM iterations (17), (18), and (16).

Proof of the $\mu-step$ EM^2 NOLC algorithm via ADMM

Similar to “Proof of the λ-step EM^{2}NOLC algorithm via ADMM” in Appendix, for the μ - step EM^{2}NOLC model (9) and its ADMM optimization problem (20) with the separable objective functions and equality constraints, the augmented Lagrangian function is denoted as.

$$
\boldsymbol{L}_\rho(\boldsymbol{\mu}, z, \boldsymbol{v}) = h(\boldsymbol{\mu}) + l(z) + (\rho/2)\|\boldsymbol{\mu} - z + \boldsymbol{v}\|_2^2 - (\rho/2)\|\boldsymbol{v}\|_2^2,
$$

with the scaled dual variables $\boldsymbol{v}(\boldsymbol{v} \in \mathbb{R}^d)$.

The ADMM updates (24), (25), and (23) of the μ - step EM^{2}NOLC algorithm are, respectively, generated from the KKT optimality conditions of $\boldsymbol{L}_\rho(\boldsymbol{\mu}, \mathbf{z}, \boldsymbol{v})$ regarding its three parameters $\boldsymbol{\mu}$, $\mathbf{z}$, and $\boldsymbol{v}$. The μ - minimization is defined as the proximal operator:

$$
\begin{aligned}
\boldsymbol{\mu}^{k+1} &= \mathop{\arg\min}_{\mu\in\mathbb{R}^d}\boldsymbol{L}_\rho(\boldsymbol{\mu}, z^k, \boldsymbol{v}^k)\\
&= \mathop{\arg\min}_{\mu\in\mathbb{R}^d}\left\{h(\mu) + (\rho/2)\left\|\boldsymbol{\mu} - z^k + \boldsymbol{v}^k\right\|_2^2\right\}.
\end{aligned}
$$

Setting the gradient of $\boldsymbol{L}_\rho(\boldsymbol{\mu}, z^k, \boldsymbol{v}^k)$ with respect to $\boldsymbol{\mu}$ to zero, we can get the analytical solution, that is the μ - update:

$$
\begin{aligned}
&\nabla_{\boldsymbol{\mu}}\boldsymbol{L}_\rho(\boldsymbol{\mu}, \mathbf{z}^k, \boldsymbol{v}^k) = 0\\
&\Rightarrow \nabla_{\boldsymbol{\mu}}\left\{h(\boldsymbol{\mu}) + (\rho/2)\left\|\boldsymbol{\mu} - \mathbf{z}^k + \boldsymbol{v}^k\right\|_2^2\right\} = 0\\
&\Rightarrow \nabla_{\boldsymbol{\mu}}\left\{(1/2)\|\mathbf{y}\odot(\boldsymbol{B}\mu - b_2\mathbf{1}_n) - \mathbf{1}_n\|_2^2 + (\rho/2)\left\|\boldsymbol{\mu} - \mathbf{z}^k + \boldsymbol{v}^k\right\|_2^2\right\} = 0\\
&\Rightarrow \boldsymbol{B}_y^T\boldsymbol{B}_y\boldsymbol{\mu} - \boldsymbol{B}_y^T(b_2\mathbf{y} + \mathbf{1}_n) + \rho\boldsymbol{\mu} - \rho\left(\mathbf{z}^k - \boldsymbol{v}^k\right) = 0\\
&\Rightarrow \boldsymbol{\mu}^{k+1} = \left(\boldsymbol{B}_y^T\boldsymbol{B}_y + \rho\boldsymbol{I}_d\right)^{-1}\left\{\boldsymbol{B}_y^T(b_2\mathbf{y} + \mathbf{1}_n) + \rho\left(\mathbf{z}^k - \boldsymbol{v}^k\right)\right\}.
\end{aligned}
$$

The z - minimization of the μ - step EM^{2}NOLC algorithm has the minimization problem with the form

$$
\begin{aligned}
\mathbf{z}^{k+1} &= \mathop{\arg\min}_{\mathbf{z}\in\mathbb{R}^d}\boldsymbol{L}_\rho(\boldsymbol{\mu}^{k+1}, \mathbf{z}, \boldsymbol{v}^k)\\
&= \mathop{\arg\min}_{\mathbf{z}\in\mathbb{R}^d}\left\{l(\mathbf{z}) + (\rho/2)\left\|\boldsymbol{\mu}^{k+1} - \mathbf{z} + \boldsymbol{v}^k\right\|_2^2\right\}.
\end{aligned}
$$

Similarly, owing to $S_0(C_\mu) = \{\mathbf{z} \in \mathbb{R}^d | \|\mathbf{z}\|_0 \leq C_\mu\}$ is a nonconvex set, we can apply the projected gradient method to the z - minimization to obtain the z - update:

$$
\begin{aligned}
\mathbf{z}^{k+1} &= \mathop{\arg\min}_{\mathbf{z}\in\mathbb{R}^d}\left\{I_{S_0(C_\mu)}(\mathbf{z}) + (\rho/2)\left\|\boldsymbol{\mu}^{k+1} - \mathbf{z} + \boldsymbol{v}^k\right\|_2^2\right\}\\
&\approx \mathop{\arg\min}_{\mathbf{z}\in S_0(C_\mu)}\left\|\boldsymbol{\mu}^{k+1} + \boldsymbol{v}^k - \mathbf{z}\right\|_2^2\\
&= \Pi_{S_0(C_\mu)}(\boldsymbol{\mu}^{k+1} + \boldsymbol{v}^k),
\end{aligned}
$$

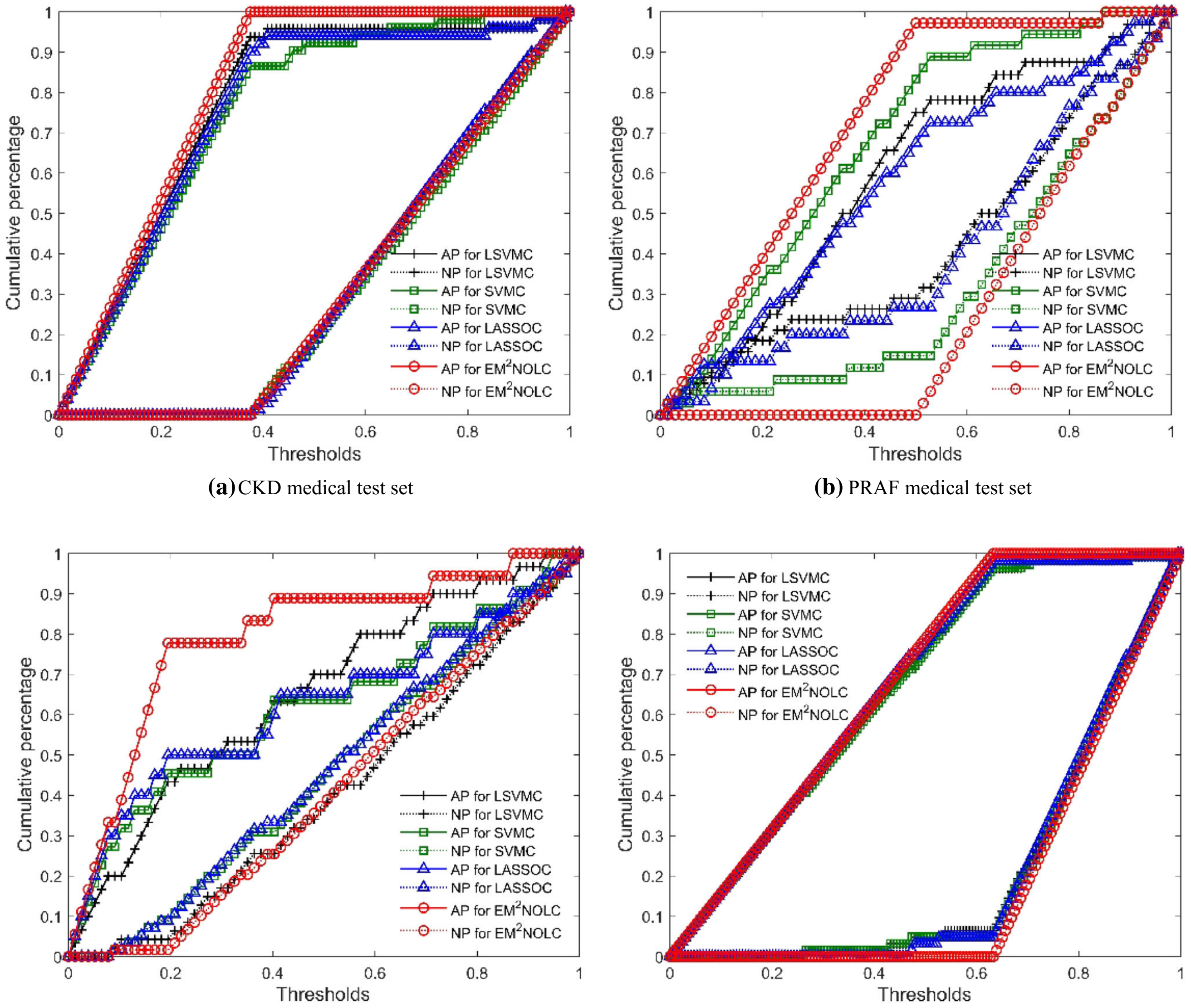
with the indicator function $I_{S_0(C_\mu)}(\mathbf{z}) = 1$ for $\mathbf{z} \in S_0(C_\mu)$ and $I_{S_0(C_\mu)}(\mathbf{z}) = 0$ for $\mathbf{z} \notin S_0(C_\mu)$. For any vector $\mathbf{z}(\mathbf{z} \in \mathbb{R}^d)$, the projection operator $\Pi_{S_0(C_\mu)}(\mathbf{z})$ can be carried out by sorting the elements of the vector $\mathbf{z}$ in descending order of their absolute values and setting all elements to zeros except the largest C_μ elements.

Finally, the v - update step can be regarded as the change of constraint residuals in the ADMM problem (20), which guarantees the convergence of the ADMM updates (24), (25), and (23).

Appendix B

KS curves in Fig. 2

See Fig. 2.



(a) CKD medical test set

(b) PRAF medical test set

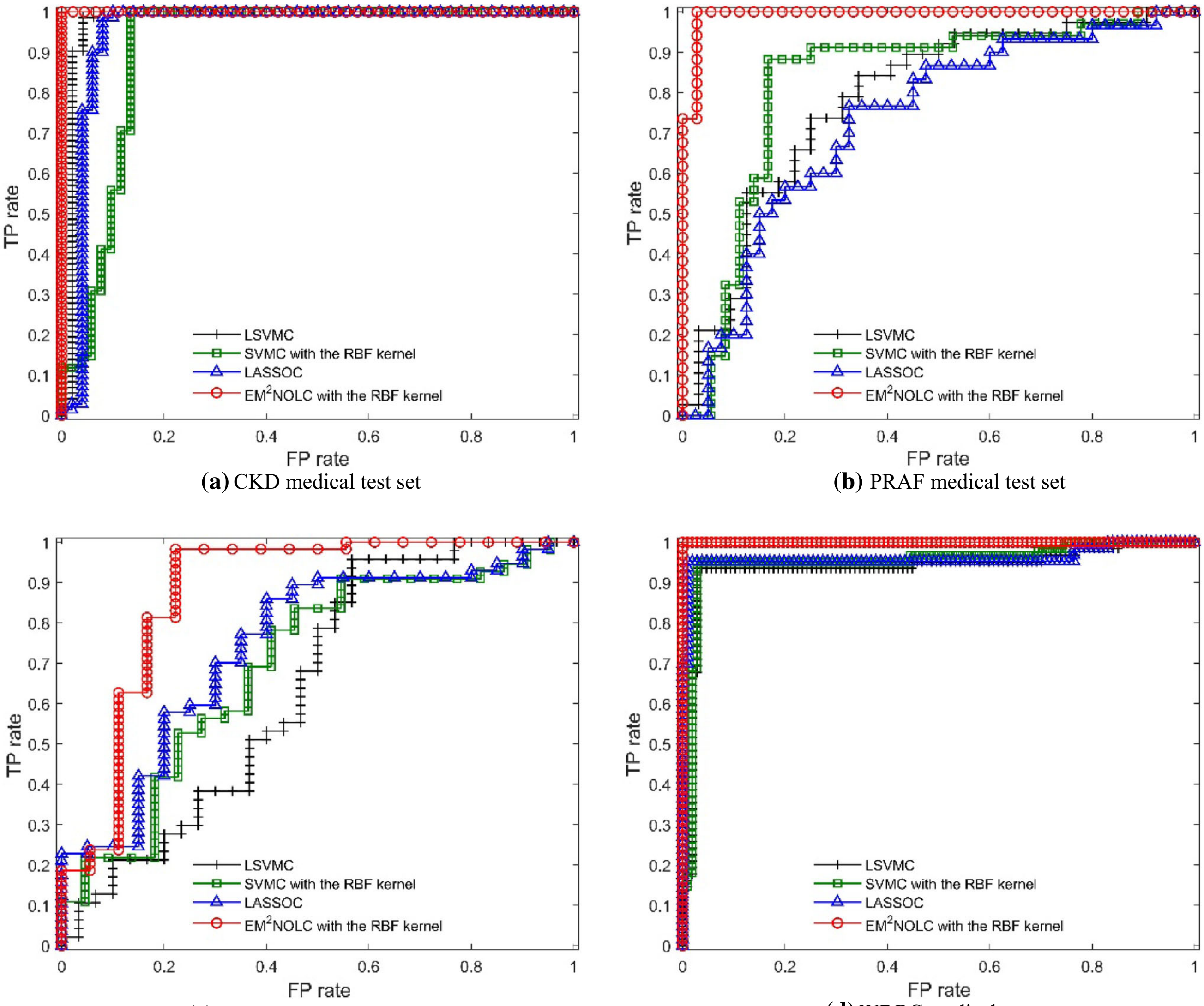
(c) SPECTFH medical test set

(d) WDBC medical test set

Fig. 2 KS curves of four ADMM classifiers on four medical test sets (the upper sold lines for AP and the lower dot lines for NP)

ROC curves in Fig. 3

See Fig. 3.



(a) CKD medical test set

(b) PRAF medical test set

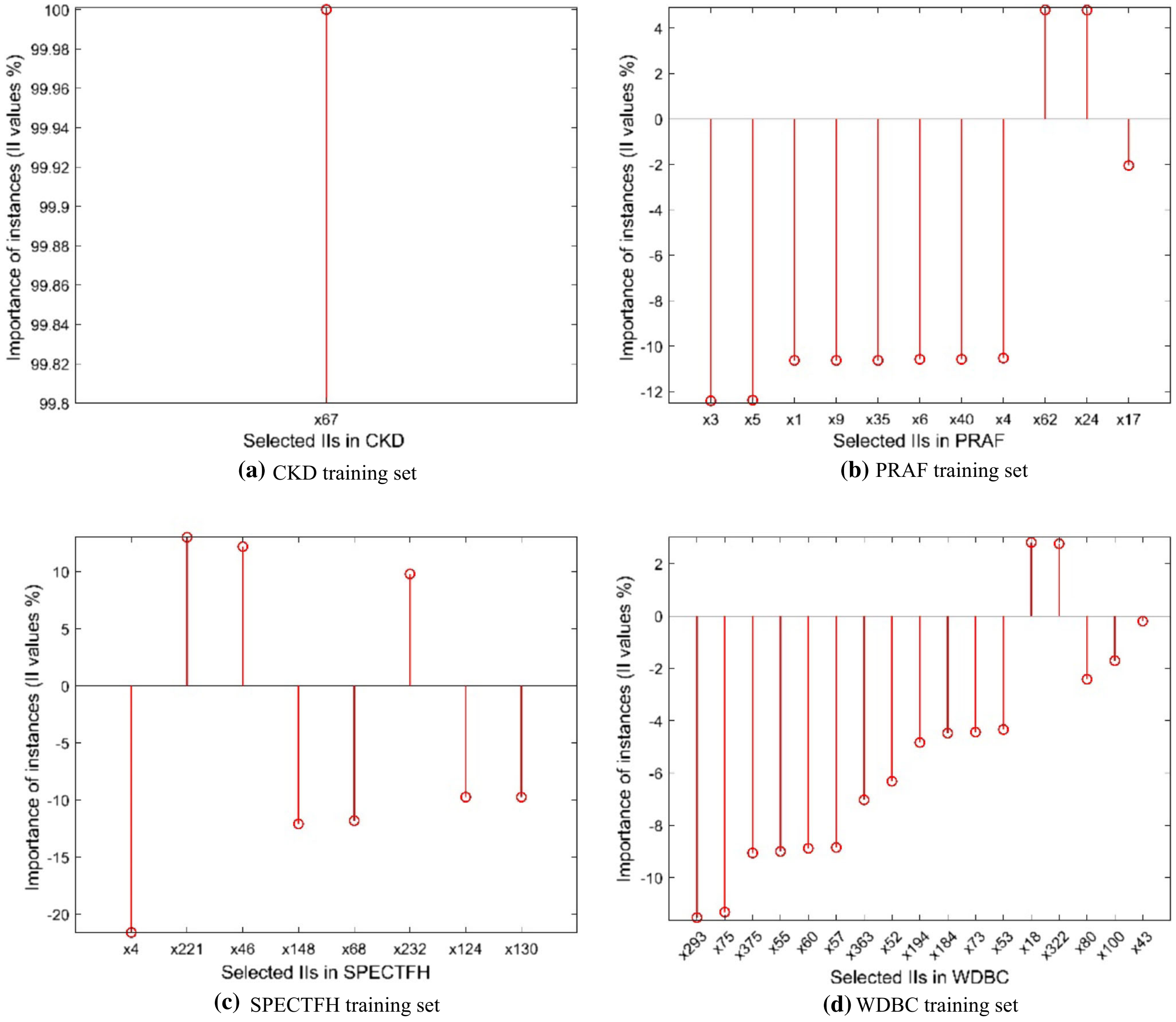
(c) SPECTFH medical test set

(d) WDBC medical test set

Fig. 3 ROC curves of four ADMM classifiers on four medical test sets

IFs for EM2NOLC in Fig. 4

See Fig. 4.



(a) CKD training set

(b) PRAF training set

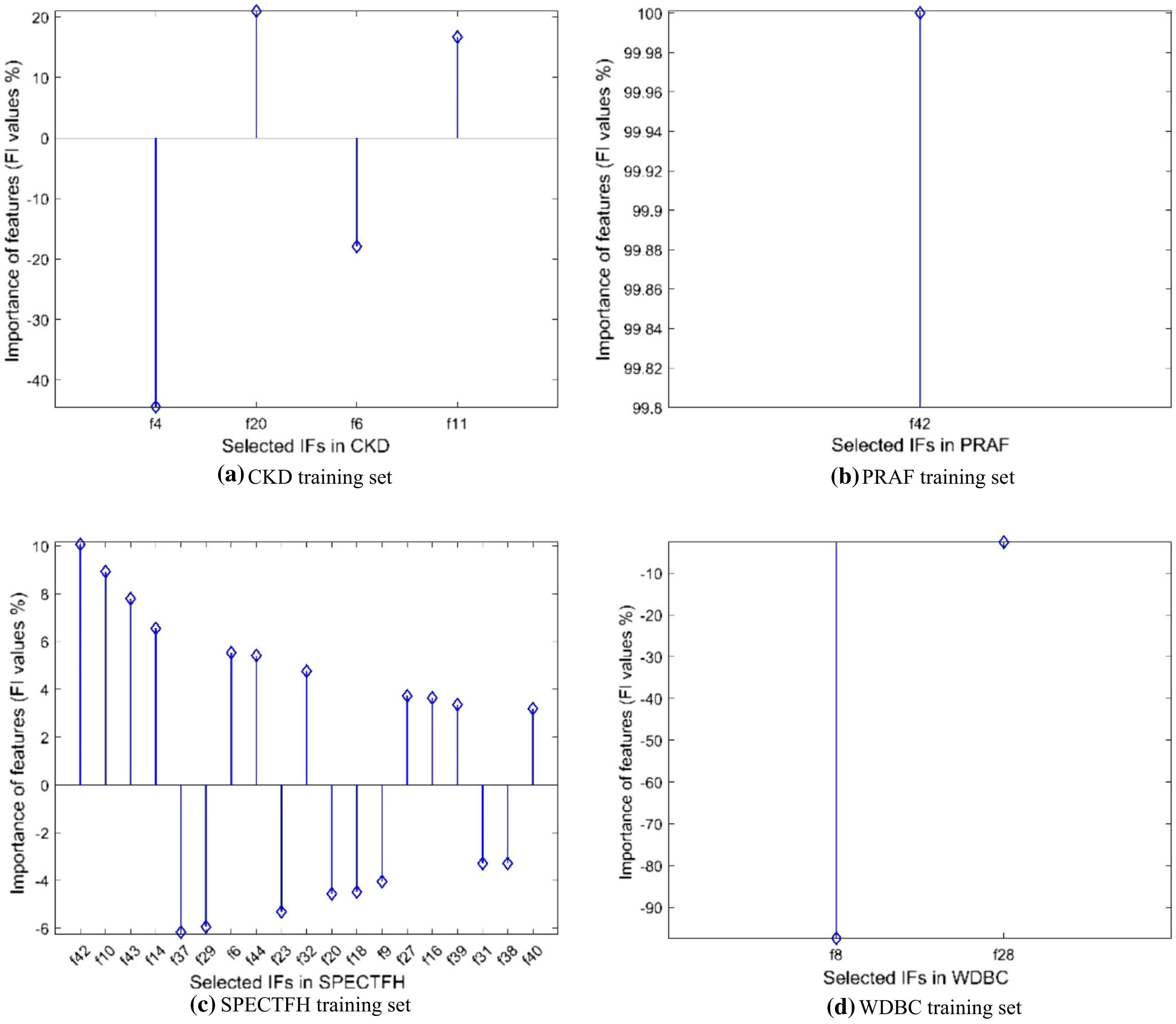
(c) SPECTFH training set

(d) WDBC training set

Fig. 4 Important instances (IIs) with their percentage values of II (%) given by EM^{2}NOLC on four training sets

IFs for EM2NOLC in Fig. 5

See Fig. 5.



(a) CKD training set

(b) PRAF training set

(c) SPECTFH training set

(d) WDBC training set

Fig. 5 Important features (IFs) with their percentage values of FI (%) given by EM^{2}NOLC on four training sets

Correlation analysis of selected features in Fig. 6

See Fig. 6.

Correlation analysis of selected features in Fig. 7

See Fig. 7.

Fig. 6 Correlation analysis of selected features across folds for the GC dataset

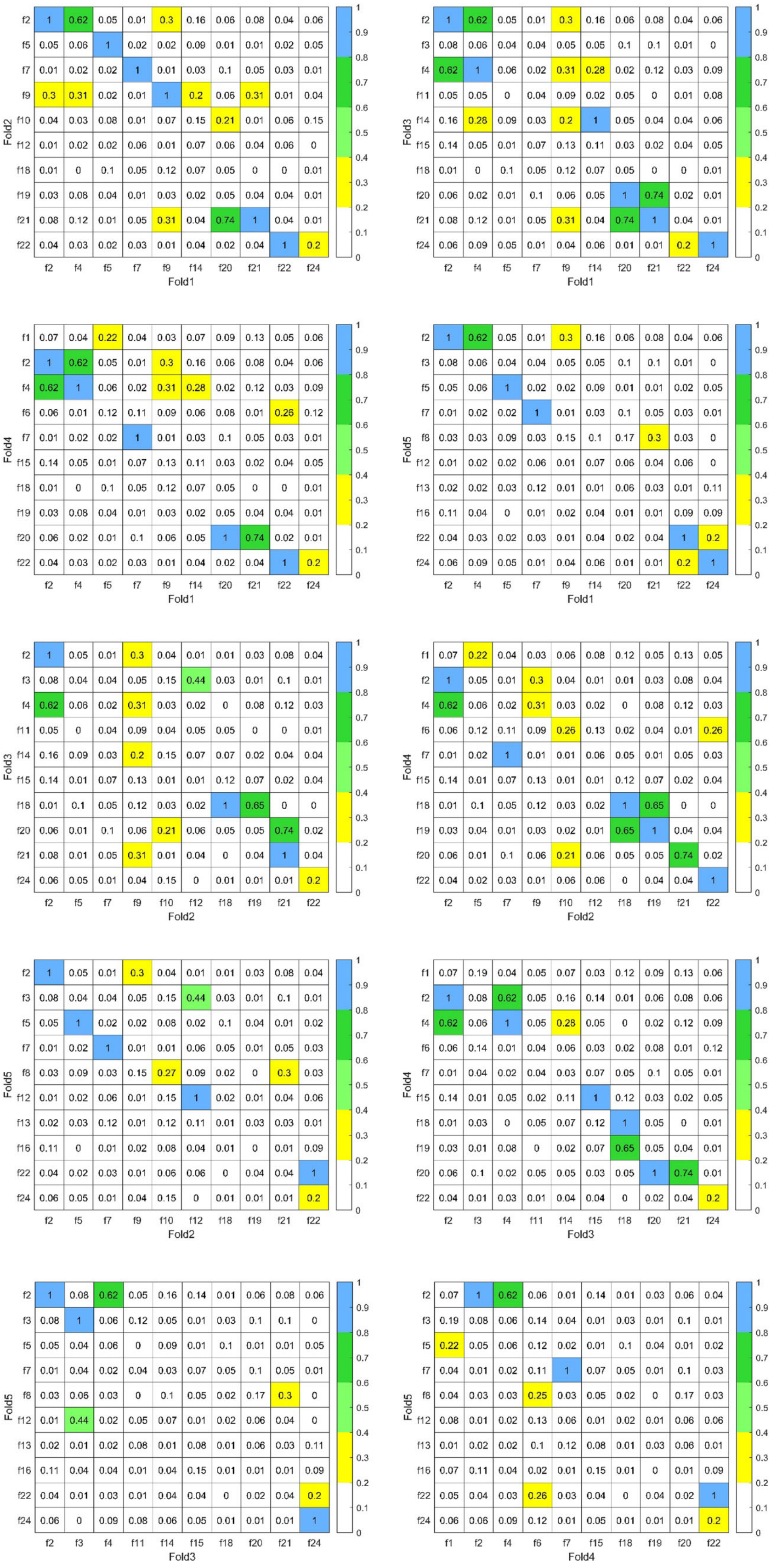
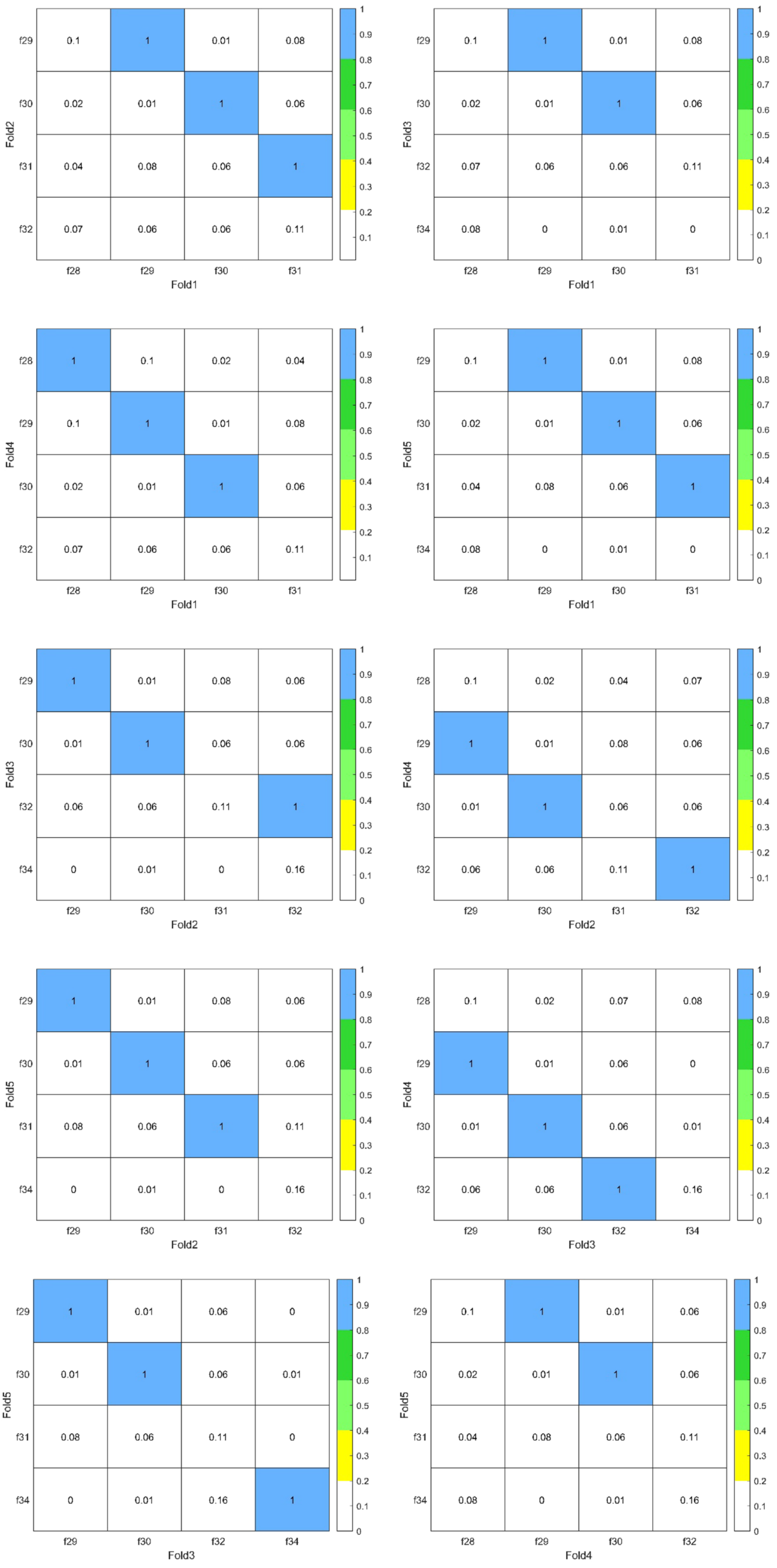


Fig. 7 Correlation analysis of selected features across folds for the MSJ dataset



Acknowledgements The authors would like to thank anonymous reviewers for their valuable comments and suggestions. This research has been partially supported by the Key Program of National Natural Science Foundation of China under grant 92046026, in part by the National Natural Science Foundation of China (No. 61877061, 71271191, 71871109, 91646204, 71701089), in part by the Jiangsu Provincial Key Research and Development Program under Grant BE2020001-3, in part by the Jiangsu Provincial Policy Guidance Program under grant BZ2020008, and in part by the High-End Foreign Experts Projects under grant G2021194011L.

Authors' contribution ZZ contributed to conceptualization, methodology, validation, formal analysis, writing—original draft, funding acquisition. JH contributed to methodology, investigation, resources, validation. JC contributed to data curation, supervision, funding acquisition. Shuqing Li contributed to validation, software, visualization. XL contributed to formal analysis, visualization, funding acquisition. KZ contributed to data curation, visualization, validation. PW contributed to formal analysis, writing—checking, resources. SY contributed to writing—review and editing, supervision, project administration.

Declarations

Conflict of interests The authors declare that they have no conflict of interests in this paper.

References

1. Sra S, Nowozin S, Wright SJ (eds) (2012) Optimization for machine learning. Mit Press, Cambridge
2. Yang X (2019) Introduction to algorithms for data mining and machine learning. Academic Press, Cambridge
3. Kantardzic M (2020) Data mining concepts, models, methods, and algorithms, 3rd edn. Wiley-IEEE Press, Hoboken
4. Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge
5. Shigeo A (2010) Support vector machines for pattern classification, 2nd edn. Springer, Berlin
6. Deng N, Tian Y, Zhang C (2012) Support vector machines: optimization-based theory, algorithms, and extensions. CRC Press, Boca Raton
7. Simeone O (2018) A brief introduction to machine learning for engineers. Found Trends Signal Process 12(3–4):200–431
8. Shawe-Taylor J, Cristianini N (2004) Kernel methods for pattern analysis. Cambridge University Press, Cambridge
9. Rakotomamonjy A, Bach FR, Canu S, Grandvalet Y (2008) SimpleMKL. J Mach Learn Res 9:2491–2521
10. Gönen M, Alpaydin E (2011) Multiple kernel learning algorithms. J Mach Learn Res 12:2211–2268
11. Gu Y, Liu T, Jia X, Benediktsson JA, Chanussot J (2016) Nonlinear multiple Kernel learning with multiple-structure-element extended morphological Profiles for hyperspectral image classification. IEEE Trans Geosci Remote Sens 54(6):3235–3247
12. Zien, A., & Ong, C. S. (2007). Multiclass multiple kernel learning. In Proceedings of the 24th international conference on Machine learning, pages 1191–1198, ACM.
13. Wang T, Zhao D, Feng Y (2013) Two-stage multiple kernel learning with multiclass kernel polarization. Knowl-Based Syst 48:10–16
14. Nazarpour A, Adibi P (2015) Two-stage multiple kernel learning for supervised dimensionality reduction. Pattern Recogn 48(5):1854–1862
15. Sonnenburg S, Rätsch G, Schäfer C, Schölkopf B (2006) Large scale multiple kernel learning. J Mach Learn Res 7:1531–1565
16. Aiolli F, Donini M (2015) EasyMKL: a scalable multiple kernel learning algorithm. Neurocomputing 169:215–224
17. Lauriola I, Gallicchio C, Aiolli F (2020) Enhancing deep neural networks via multiple kernel learning. Pattern Recogn 101:107194
18. Zhang Z, Gao G, Yao T, He J, Tian Y (2020) An interpretable regression approach based on bi-sparse optimization. Appl Intell 50(11):4117–4142
19. Bach F, Jenatton R, Mairal J, Obozinski G (2011) Optimization with sparsity-inducing penalties. Found Trends Mach Learn 4(1):1–106
20. Rish I, Grabarnik GY (2014) Sparse modeling: theory, algorithms, and applications. Chapman & Hall/CRC Press, Boca Raton
21. Gregorova M (2019) Sparse learning for variable selection with structures and nonlinearities. Doctoral dissertation, Geneve
22. Jain P, Kar P (2017) Non-convex optimization for machine learning. Found Trends Mach Learn 10(3–4):142–336
23. Weston J, Elisseeff A, Schölkopf B, Tipping M (2003) Use of the zero-norm with linear models and kernel methods. J Mach Learn Res 3:1439–1461
24. Huang K, Zheng D, Sun J, Hotta Y, Fujimoto K, Naoi S (2010) Sparse learning for support vector classification. Pattern Recogn Lett 31(13):1944–1951
25. Zhu J, Rosset S, Tibshirani R, Hastie TJ (2004) 1-norm support vector machines. In Advances in neural information processing systems, pages 49–56
26. Wang L, Shen X (2007) On L1-Norm Multiclass Support Vector Machines. J Am Stat Assoc 102(478):583–594
27. Chapelle O, Keerthi SS (2008) Multi-class feature selection with support vector machines. In Proceedings of the American statistical association
28. Mairal J, Bach F, Ponce J (2012) Sparse modeling for image and vision processing. Found Trends Comput Graph Vis 8(2–3):85–283
29. Tibshirani R (1996) Regression shrinkage and selection via the lasso. J R Stat Soc Ser B (Methodological) 58:267–288
30. Yamada M, Jitkrittum W, Sigal L, Xing EP, Sugiyama M (2014) High-dimensional feature selection by feature-wise kernelized lasso. Neural Comput 26(1):185–207
31. Sjöstrand K, Clemmensen LH, Larsen R, Einarsson G, Ersbøll BK (2018) Spasm: a matlab toolbox for sparse statistical modeling. J Stat Softw 84(10):1–37
32. Weston J, Mukherjee S, Chapelle O, Pontil M, Poggio T, Vapnik V (2000) Feature selection for SVMs
33. Parikh N, Boyd S (2013) Proximal algorithms. Found Trends Optim 1(3):123–231
34. Boyd S, Parikh N, Chu E, Peleato B, Eckstein J (2010) Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends Mach Learn 3(1):1–122
35. Beck A (2017) First-order methods in optimization. Mathematical optimization society and the society for industrial and applied mathematics, Philadelphia, PA 19104–2688 USA
36. Gallier J, Quaintance J (2019) Fundamentals of optimization theory with applications to machine learning. University of Pennsylvania, Philadelphia
37. Theodoridis S (2020) Machine learning a Bayesian and optimization perspective, 2nd edn. Academic Press, Elsevier

38. Shalev-Shwartz S, Ben-David S (2014) Understanding machine learning: from theory to algorithms. Cambridge University Press, Cambridge
39. Bottou L, Curtis EF, Nocedal J (2018) Optimization methods for large-scale machine learning. SIAM Rev 60(2):223–311
40. Shalev-Shwartz S (2011) Online learning and online convex optimization. Found Trends Mach Learn 4(2):107–194
41. Hazan E (2015) Introduction to online convex optimization. Found Trends Optim 2(3–4):157–325
42. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. The MIT Press, Cambridge
43. Charniak E (2019) Introduction to deep learning. The MIT Press, Cambridge
44. Cao J, Wang Y, He J, Liang W, Tao H, Zhu G (2021) Predicting grain losses and waste rate along the entire chain: a multitask multigated recurrent unit autoencoder based method. IEEE Trans Industr Inform 17(6):4390–4400
45. Hall, P. & Gill, N. (2019). An Introduction to Machine Learning Interpretability, An Applied Perspective on Fairness, Accountability, Transparency, and Explainable AI, 2nd Edition. O'Reilly Media, Inc.
46. Murdoch WJ, Singh C, Kumbier K, Abbasi-Asl R, Yu B (2019) Interpretable machine learning: definitions, methods, and applications. PNAS 116(44):22071–22080
47. Molnar C (2021). Interpretable machine learning, a guide for making black box models explainable. Leanpub.com
48. Hastie T, Tibshirani R, Wainwright M (2015) Statistical learning with sparsity: the lasso and generalizations. CRC Press, Boca Raton
49. Suykens JA, Vandewalle J, Moor BD (2001) Optimal control by least squares support vector machines. Neural Netw 14(1):23–35
50. Xanthopoulos P, Pardalos PM, Trafalis TB (2012) Robust data mining. Springer Science & Business Media, Berlin
51. Boyd S, Vandenberghe L (2018) Introduction to applied linear algebra vectors, matrices, and least squares. Cambridge University Press, Cambridge
52. Dua D, Graff C (2019) UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science
53. Hand DJ (2009) Measuring classifier performance: a coherent alternative to the area under the ROC curve. Mach Learn 77:103–123
54. Matlab, http://www.mathworks.com
55. https://web.stanford.edu/∼boyd/index.html

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.