Front. Comput. Sci., 2019, 13(4): 813–827
https://doi.org/10.1007/s11704-018-7402-8

RESEARCH ARTICLE

A maximum margin clustering algorithm based on indefinite kernels

Hui XUE (✉)[1,2], Sen LI[1,2], Xiaohong CHEN[3], Yunyun WANG[4]

1 School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
2 Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing 210096, China
3 College of Science, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
4 Department of Computer Science and Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210046, China

© Higher Education Press and Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract Indefinite kernels have attracted more and more attentions in machine learning due to its wider application scope than usual positive definite kernels. However, the research about indefinite kernel clustering is relatively scarce. Furthermore, existing clustering methods are mainly designed based on positive definite kernels which are incapable in indefinite kernel scenarios. In this paper, we propose a novel indefinite kernel clustering algorithm termed as indefinite kernel maximum margin clustering (IKMMC) based on the state-of-the-art maximum margin clustering (MMC) model. IKMMC tries to find a proxy positive definite kernel to approximate the original indefinite one and thus embeds a new F-norm regularizer in the objective function to measure the diversity of the two kernels, which can be further optimized by an iterative approach. Concretely, at each iteration, given a set of initial class labels, IKMMC firstly transforms the clustering problem into a classification one solved by indefinite kernel support vector machine (IKSVM) with an extra class balance constraint and then the obtained prediction labels will be used as the new input class labels at next iteration until the error rate of prediction is smaller than a prespecified tolerance. Finally, IKMMC utilizes the prediction labels at the last iteration as the expected indices of clusters. Moreover, we further extend IKMMC from binary clustering problems to more complex multi-class scenarios. Experimental results have shown the superiority of our algorithms.

Keywords indefinite kernel, maximum margin clustering, support vector machine, kernel method

Received November 15, 2017; accepted April 2, 2018

E-mail: hxue@seu.edu.cn

1 Introduction

Kernel method is one of the most powerful techniques in machine learning. It works by embedding original data into a high-dimensional feature space where the embedding is defined implicitly through a kernel function, in order to transform the original non-linear learning problems into linear ones. In the traditional statistical learning theory, kernel functions are required to be positive definite (PD) satisfying the mercer condition [1] and the corresponding kernel matrix should be positive semi-definite (PSD) to ensure that original data can be mapped into a reproducing kernel hilbert space (RKHS) [2]. As a result, if the original learning problems are convex, they can still preserve their convexity after kernelization and achieve a global optimal solution in the RKHS.

However, such requirement seems to be too strict and difficult to be satisfied in practical problems [3–8]. On one hand, standard PD kernels are not suitable in many situations, such as suboptimal optimization procedures for measure deriva-

tion [9], partial projections or occlusions [10], and context-dependent alignments or object comparisons [11]. On the other hand, several applications have shown that non-PD kernels can result in better performance than PD ones [12]. For example, in face recognition, Liu [13] utilized a non-PD fractional power polynomial kernel in the kernel principal component analysis (KPCA), which can perform much better than the PD polynomial kernels.

Recently, a series of kernel methods have been proposed to generalize the kernels from PD to non-PD (that is, indefinite kernel) scenarios. The simplest method is spectrum transformation, which generates a new PSD kernel matrix by directly transforming the spectrum of the indefinite kernel matrix. The corresponding representative algorithms include: "*Clip*" which neglects the negative eigenvalues [14,15], "*Flip*" which flips the sign of the negative eigenvalues [16] and "*Shift*" which shifts all the eigenvalues by a positive constant [17]. Although these algorithms are quite simple, the valuable information involved by negative eigenvalues is artificially lost.

PD kernel proxy is another more sophisticated method, which was firstly proposed by Luss and d'Aspremont [18]. They considered an indefinite kernel as a noisy observation of some unknown PD kernel (proxy kernel) and the corresponding objective function is convex. In order to facilitate the calculation of the gradient, Luss and d'Aspremont [18] quadratically smoothed the objective function. Then they proposed the projected gradient algorithm and the cutting plane algorithm. Waldspurger reformulated the objective function as a semi-infinite quadratically constrained linear problem (SIQCLP) [19], and solved the problem iteratively to find a global optimum solution [20]. Auslender [21] and Chen et al. [22] further expressed the function as a second order cone programming (SOCP) problem. Gu and Guo [23] firstly expressed KPCA as a general kernel transformation framework and then incorporated the indefinite kernel support vector machine (IKSVM) into the framework to formulate a joint max-min optimization model.

Several other algorithms are designed to solve the non-convex indefinite kernel problem directly. For example, Lin and Lin [24] proposed an SMO-type method to find a stationary point in the non-convex dual formulation of SVM based on a non-PD sigmoid kernel. Haasdonk gave a geometric interpretation of IKSVM model and the corresponding optimization problem can be reformulated as finding the minimum distance between two convex hulls in some pseudo-Euclidean space [25]. Ong et al. further extended indefinite kernels to the reproducing kernel krein space (RKKS) [26-28]. Xu et al. [29] directly solved a primal form of IKSVM with difference of convex functions programming.

In the past few years, these indefinite kernel algorithms have shown much better performance in classification problems. Xue et al. [30] further extended indefinite kernels into feature selection problems and proposed a multiple indefinite kernel feature selection algorithm (MIK-FS). Experimental results have shown that MIK-FS is superior to some related state-of-the-art algorithms in both feature selection and classification performance.

Inspired by these successful applications of indefinite kernels, we naturally consider whether indefinite kernels can also perform better in clustering problems. However, so far, the research about indefinite kernel clustering is still relatively scarce. Consequently, here we will focus on this problem directly.

In view of the excellent performance of indefinite kernel classification methods with PD kernel proxy, we expect to utilize these methods to solve clustering problems and thus propose a novel algorithm termed as indefinite kernel maximum margin clustering (IKMMC) based on the state-of-the-art maximum margin clustering model (MMC) [31–37]. Different from usual MMC algorithms with PD kernels, IKMMC embeds a F-norm regularizer into the model used to measure the difference between the indefinite kernel and proxy PD kernel, and further characterizes the corresponding model as an indefinite kernel classification problem optimized by an iterative approach.

Concretely, at each iteration, given a set of initial class labels, IKMMC model can be firstly transformed into a classification problem as an IKSVM with an extra class balance constraint, and then formulated as a semi-infinite programming (SIP) [38] to solve. The obtained prediction labels will be used as the new input class labels at next iteration and the whole iteration procedure continues until the error rate of prediction is smaller than a pre-specified tolerance. Finally, the prediction labels at the last iteration are obtained as the expected indices of clusters in IKMMC.

The rest of the paper is organized as follows. In Section 2, we give a brief review on MMC model. In Section 3, IKMMC model is proposed and the corresponding algorithm is presented. In Section 4, we extend IKMMC from two-class to the multi-class scenarios. Systematically experimental comparisons with some relative algorithms are reported in Section 5. Some conclusions are drawn in Section 6.

2 Maximum margin clustering (MMC)

Traditional MMC model is based on PD kernels, which aims

to extend the maximum margin principle of SVM to the unsupervised clustering scenarios.

In two-class clustering settings, given a data set $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \in \boldsymbol{R}^d$, MMC seeks the hyperplane as well as the predictive labels by solving the following optimization problem:

$$\begin{aligned} \min_{y\in\{\pm1\}^n} \min_{\boldsymbol{w},b,\xi} \ & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i, \\ s.t. \quad & y_i(\boldsymbol{w}^{\mathrm{T}}\phi(x_i) + b) \geqslant 1 - \xi_i, \quad \forall i \in \{1, \ldots, n\}, \\ & \xi_i \geqslant 0, \forall i \in \{1, \ldots, n\}, \\ & -l \leqslant \sum_{i=1}^{n} y_i \leqslant l, \end{aligned} \tag{1}$$

where $\boldsymbol{\xi} = [\xi_1, \xi_2, \ldots, \xi_n]^{\mathrm{T}}$ is the vector of slack variables, $C > 0$ is the regularization parameter, and ϕ is a possibly nonlinear feature mapping.

Since the class labels are unknown, a trivially "optimal" solution is to assign all samples to the same class which would result in an infinite margin. In order to prevent this meaningless solution, Xu et al. [31] introduced the last balance constraint in Eq. (1) where $l > 0$ is a constant controlling the class imbalance.

Obviously, Eq. (1) is an integer programming which is more difficult to solve than the quadratic programming. According to [33], Eq. (1) can be slightly relaxed as the following formulation:

$$\begin{aligned} \min_{\boldsymbol{w},b,\xi} \ & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i, \\ s.t. \quad & |\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}_i) + b| \geqslant 1 - \xi_i, \quad \forall i \in \{1, \ldots, n\}, \\ & \xi_i \geqslant 0, \quad \forall i \in \{1, \ldots, n\}, \\ & -l \leqslant \sum_{i=1}^{n} [\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}_i) + b] \leqslant l, \end{aligned} \tag{2}$$

where the label vector $\boldsymbol{y}$ is computed through $y_i = \mathrm{sign}(\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}_i) + b)$.

As can be seen, Eq. (2) is a non-convex programming due to the first n non-convex constraints which is relatively difficult to tackle. But during the past decade, a lot of MMC algorithms have been proposed to solve this non-convex objective function and achieved good performance which can be roughly divided into two categories. One category is to utilize concave convex constrained programming (CCCP) [39] and cutting plane algorithm [40] to solve the non-convex objective function directly. The other category is to transform the objective function into a series of standard SVM classification problems so as to utilize the classifiers to solve clustering, such as iterSVM [32].

3 Two-class indefinite kernel maximum margin clustering

In this section, we present a novel algorithm called as indefinite kernel maximum margin clustering (IKMMC) based on the traditional MMC model in two-class clustering scenarios.

3.1 Model construction

IKMMC considers the indefinite kernel matrix as a noisy observation of some unknown PSD one (proxy kernel) and further embeds a new F-norm regularizer in the objective function to measure the diversity of the two kernels. Computationally, two-class IKMMC solves the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{K}} \ & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i + \rho\|\boldsymbol{K} - \boldsymbol{K}_0\|_F^2, \\ s.t. \quad & |\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}_i) + b| \geqslant 1 - \xi_i, \quad \forall i \in \{1, \ldots, n\}, \\ & \xi_i \geqslant 0, \quad \forall i \in \{1, \ldots, n\}, \\ & -l \leqslant \sum_{i=1}^{n} [\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}_i) + b] \leqslant l, \\ & \boldsymbol{K} \succeq 0 \end{aligned} \tag{3}$$

where $\boldsymbol{K}_0$ is a pre-specified indefinite kernel matrix, $\boldsymbol{K}$ is the unknown proxy kernel matrix. The parameter $\rho > 0$ controls the magnitude of the penalty on the distance between $\boldsymbol{K}$ and $\boldsymbol{K}_0$, and $\|\cdot\|_F^2$ denotes the Frobenius norm of a matrix. $\phi(\boldsymbol{x}_i)$ is the mapping induced by the proxy kernel matrix.

The last constraint ensures the proxy kernel matrix is PSD. Moreover, minimizing the F-norm of $\boldsymbol{K}$ and $\boldsymbol{K}_0$ guarantees the proxy kernel matrix to be as close as possible to the original indefinite one. Obviously, the objective function in Eq. (3) is also non-convex due to the first n constraints and thus has no global optimal solution. However, we can reach a stable point by optimizing Eq. (3) alternatively.

3.2 Optimization algorithm

As mentioned above, we aim to utilize indefinite kernel classification methods to direct indefinite kernel clustering. Therefore, we adopt an iterative algorithm to solve Eq. (3) by decomposing it into a series of convex IKSVM problems referring to iterSVM [32].

Concretely, at iteration $t + 1$, we use the prediction labels at last iteration as the input labels. Therefore, we have the

following objective function:

$$\begin{aligned}
&\min_{\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{K}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i + \rho\|\boldsymbol{K}-\boldsymbol{K}_0\|_F^2,\\
&s.t. \quad z_i^{(t)}(\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}_i)+b) \geqslant 1-\xi_i, \quad \forall i \in \{1,\ldots,n\},\\
&\qquad \xi_i \geqslant 0, \quad \forall i \in \{1,\ldots,n\},\\
&\qquad -l \leqslant \sum_{i=1}^{n}[\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}_i)+b] \leqslant l,\\
&\qquad \boldsymbol{K} \succeq 0,
\end{aligned} \tag{4}$$

where $z_i^{(t)}$ is the output label vector at iteration t computed as

$$z_i^{(t)} = \mathrm{sign}(\boldsymbol{w}^{(t)\mathrm{T}}\phi^{(t)}(\boldsymbol{x}_i)+b^{(t)}).$$

Obviously, Eq. (4) is convex in $\boldsymbol{w}$, b and $\boldsymbol{K}$, and thus has a global optimal solution. Similarly to IKSVM, we can write the dual of Eq. (4) in $\boldsymbol{w}$, b and $\boldsymbol{\xi}$ as follows:

$$\begin{aligned}
&\min_{\boldsymbol{K}} \max_{\alpha,\lambda,\gamma} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j z_i^{(t)}z_j^{(t)}\phi^{\mathrm{T}}(\boldsymbol{x}_i)\phi(\boldsymbol{x}_j)\\
&\quad -(\gamma-\lambda)\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i z_i^{(t)}\phi^{\mathrm{T}}(\boldsymbol{x}_i)\phi(\boldsymbol{x}_j)\\
&\quad -\frac{1}{2}(\gamma-\lambda)^2\sum_{i=1}^{n}\sum_{j=1}^{n}\phi^{\mathrm{T}}(\boldsymbol{x}_i)\phi(\boldsymbol{x}_j)\\
&\quad -(\gamma+\lambda)l + \rho\|\boldsymbol{K}-\boldsymbol{K}_0\|_F^2,\\
&\quad s.t. \quad n(\lambda-\gamma) = \sum_{i=1}^{n}\alpha_i z_i^{(t)},\\
&\quad 0 \leqslant \alpha_i \leqslant C, \quad \forall i \in \{1,\ldots,n\},\\
&\quad \lambda \geqslant 0, \quad \gamma \geqslant 0,\\
&\quad \boldsymbol{K} \succeq 0.
\end{aligned} \tag{5}$$

For simplicity, we denote the objective function in Eq. (5) as:

$$\begin{aligned}
S(\boldsymbol{\alpha},\lambda,\gamma,\boldsymbol{K}) = &\sum_{i=1}^{n}\alpha_i - 1/2\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j z_i^{(t)}z_j^{(t)}\phi^{\mathrm{T}}(\boldsymbol{x}_i)\phi(\boldsymbol{x}_j)\\
&-(\gamma-\lambda)\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i z_i^{(t)}\phi^{\mathrm{T}}(\boldsymbol{x}_i)\phi(\boldsymbol{x}_j)\\
&-1/2(\gamma-\lambda)^2\sum_{i=1}^{n}\sum_{j=1}^{n}\phi^{\mathrm{T}}(\boldsymbol{x}_i)\phi(\boldsymbol{x}_j)\\
&-(\gamma+\lambda)l + \rho\|\boldsymbol{K}-\boldsymbol{K}_0\|_F^2.
\end{aligned}$$

Obviously, the optimal solution of Eq. (5) is a saddle point for the function $S(\boldsymbol{\alpha},\lambda,\gamma,\boldsymbol{K})$ subject to the constraints. Suppose $(\boldsymbol{\alpha}^*,\lambda^*,\gamma^*,\boldsymbol{K}^*)$ is the optimal solution. For any feasible solution, the following inequality holds:

$$S(\boldsymbol{\alpha},\lambda,\gamma,\boldsymbol{K}^*) \leqslant S(\boldsymbol{\alpha}^*,\lambda^*,\gamma^*,\boldsymbol{K}^*) \leqslant S(\boldsymbol{\alpha}^*,\lambda^*,\gamma^*,\boldsymbol{K}).$$

According to [20], Eq. (5) can be reformulated into a semi-infinite quadratically constrained linear program (SIQCLP) problem as follows:

$$\begin{aligned}
&\max_{\alpha,\lambda,\gamma,d} d\\
&s.t. \quad n(\lambda-\gamma) = \sum_{i=1}^{n}\alpha_i z_i^{(t)},\\
&\qquad 0 \leqslant \alpha_i \leqslant C, \quad \forall i \in \{1,\ldots,n\},\\
&\qquad \lambda \geqslant 0, \quad \gamma \geqslant 0,\\
&\qquad d \leqslant S(\boldsymbol{\alpha},\lambda,\gamma,\boldsymbol{K}), \quad \forall \boldsymbol{K} \succeq 0.
\end{aligned} \tag{6}$$

Specifically, the last constraint in Eq. (6) can be reformulated as

$$d \leqslant \min_{\boldsymbol{K}\succeq 0} S(\boldsymbol{\alpha},\lambda,\gamma,\boldsymbol{K}),$$

thus the saddle point can be found when we maximize d in Eq. (6) subject to the constraints.

Note that the number of $\boldsymbol{K} \succeq 0$ satisfying the last constraint in Eq. (6) is infinite. Consequently, Eq. (6) can not be optimized directly. However, we can approach the optimum by optimizing the variables with a restricted subset of the constraints and then update the constraint subset based on the obtained suboptimal solution. For a constraint subset, we have the following optimization problem [20]:

$$\begin{aligned}
&\max_{\alpha,\lambda,\gamma,d} d\\
&s.t. \quad n(\lambda-\gamma) = \sum_{i=1}^{n}\alpha_i z_i^{(t)},\\
&\qquad 0 \leqslant \alpha_i \leqslant C, \quad \forall i \in \{1,\ldots,n\},\\
&\qquad \lambda \geqslant 0, \quad \gamma \geqslant 0,\\
&\qquad d \leqslant S(\boldsymbol{\alpha},\lambda,\gamma,\boldsymbol{K}_j), \quad j \in \{1,\ldots,p\},
\end{aligned} \tag{7}$$

Then the solution to Eq. (6) can be found by optimizing Eq. (7) iteratively. At each iteration, a PD kernel $\boldsymbol{K}^*$ with the maximum violation is selected to be added to the last constraint subset in Eq. (7). With the number of $\boldsymbol{K} \succeq 0$ in Eq. (7) increasing, the solution of Eq. (7) can be infinitely close to that of Eq. (6).

In fact, the most violated matrix $\boldsymbol{K}^*$ is the one minimizes $S(\boldsymbol{\alpha},\lambda,\gamma,\boldsymbol{K})$, which can be computed according to Theorem 1.

Theorem 1 For $\boldsymbol{K}^* = \arg\min_{\boldsymbol{K}\succeq 0} S(\boldsymbol{\alpha},\lambda,\gamma,\boldsymbol{K})$, the optimal $\boldsymbol{K}^*$ is given by

$$\begin{aligned}
\boldsymbol{K}^* = (\boldsymbol{K}_0 &+ \frac{1}{4\rho}(\boldsymbol{Z}^{(t)}\boldsymbol{\alpha})(\boldsymbol{Z}^{(t)}\boldsymbol{\alpha})^{\mathrm{T}} + \frac{\gamma-\lambda}{2\rho}\boldsymbol{Z}^{(t)}\boldsymbol{\alpha}\cdot I_{n\times 1}^{\mathrm{T}}\\
&+\frac{(\gamma-\lambda)^2}{4\rho}\cdot I_{n\times n})_+,
\end{aligned} \tag{8}$$

where $\boldsymbol{Z}^{(t)} = \mathrm{diag}(z_1^{(t)}, \ldots, z_n^{(t)})$, $1_{n\times 1}$ is a vector of all ones with the length n, $\boldsymbol{U}_+ = \sum_i \max(0, \lambda_i)\boldsymbol{u}_i\boldsymbol{u}_i^{\mathrm{T}}$, λ_i and $\boldsymbol{u}_i$ denote the ith eigenvalue and eigenvector [18].

Proof For simplicity, $S(\boldsymbol{\alpha}, \lambda, \gamma, \boldsymbol{K})$ can be reformulated as

$$\begin{aligned}\sum_{i=1}^{n}\alpha_i - 1/2Tr(\boldsymbol{K}(\boldsymbol{Z}^{(t)}\boldsymbol{\alpha})(\boldsymbol{Z}^{(t)}\boldsymbol{\alpha})^{\mathrm{T}})\\ -(\gamma-\lambda)Tr(\boldsymbol{K}(\boldsymbol{Z}^{(t)}\boldsymbol{\alpha}\cdot 1_{n\times 1}^{\mathrm{T}}))\\ -1/2(\gamma-\lambda)^2 Tr(\boldsymbol{K}\cdot 1_{n\times n})\\ -(\gamma+\lambda)l + \rho\|\boldsymbol{K}-\boldsymbol{K}_0\|_F^2.\end{aligned}$$

For a fixed $(\boldsymbol{\alpha}, \lambda, \gamma)$, the optimal $\boldsymbol{K}^*$ is the solution of the optimization problem: $\min_{\boldsymbol{K}\geq 0} S(\boldsymbol{\alpha}, \lambda, \gamma, \boldsymbol{K})$. Specially, $\|\boldsymbol{K}-\boldsymbol{K}_0\|_F^2$ can be reformulated as $Tr((\boldsymbol{K}-\boldsymbol{K}_0)^{\mathrm{T}}(\boldsymbol{K}-\boldsymbol{K}_0))$. So, $\min_{\boldsymbol{K}\geq 0} S(\boldsymbol{\alpha}, \lambda, \gamma, \boldsymbol{K})$ can be written as:

$$\begin{aligned}\min_{\{\boldsymbol{K}\geq 0\}} \rho\cdot Tr(\boldsymbol{K}^{\mathrm{T}}\boldsymbol{K} - 2\boldsymbol{K}^{\mathrm{T}}(\boldsymbol{K}_0 + \frac{1}{4\rho}(\boldsymbol{Z}^{(t)}\boldsymbol{\alpha})(\boldsymbol{Z}^{(t)}\boldsymbol{\alpha})^{\mathrm{T}}\\ +\frac{\gamma-\lambda}{2\rho}\boldsymbol{Z}^{(t)}\boldsymbol{\alpha}\cdot 1_{n\times 1}^{\mathrm{T}} + \frac{(\gamma-\lambda)^2}{4\rho}\cdot 1_{n\times n}) + \boldsymbol{K}_0^{\mathrm{T}}\boldsymbol{K}_0).\end{aligned}$$

Notice that $(\alpha, \lambda, \gamma)$ is a constant term for the above problem, we can replace $\rho Tr(\boldsymbol{K}_0^{\mathrm{T}}\boldsymbol{K}_0)$ by

$$\begin{aligned}\rho Tr((\boldsymbol{K}_0 + \frac{1}{4\rho}(\boldsymbol{Z}^{(t)}\boldsymbol{\alpha})(\boldsymbol{Z}^{(t)}\boldsymbol{\alpha})^{\mathrm{T}} + \frac{\gamma-\lambda}{2\rho}\boldsymbol{Z}^{(t)}\boldsymbol{\alpha}\cdot 1_{n\times 1}^{\mathrm{T}}\\ +\frac{(\gamma-\lambda)^2}{4\rho}\cdot 1_{n\times n})^{\mathrm{T}} * (\boldsymbol{K}_0 + \frac{1}{4\rho}(\boldsymbol{Z}^{(t)}\boldsymbol{\alpha})(\boldsymbol{Z}^{(t)}\boldsymbol{\alpha})^{\mathrm{T}}\\ +\frac{\gamma-\lambda}{2\rho}\boldsymbol{Z}^{(t)}\boldsymbol{\alpha}\cdot 1_{n\times 1}^{\mathrm{T}} + \frac{(\gamma-\lambda)^2}{4\rho}\cdot 1_{n\times n})),\end{aligned}$$

resulting the following optimization problem about $\boldsymbol{K}$:

$$\begin{aligned}\min_{\{\boldsymbol{K}\geq 0\}} \|\boldsymbol{K} - (\boldsymbol{K}_0 + \frac{1}{4\rho}(\boldsymbol{Z}^{(t)}\boldsymbol{\alpha})(\boldsymbol{Z}^{(t)}\boldsymbol{\alpha})^{\mathrm{T}} + \frac{\gamma-\lambda}{2\rho}\boldsymbol{Z}^{(t)}\boldsymbol{\alpha}\cdot 1_{n\times 1}^{\mathrm{T}}\\ +\frac{(\gamma-\lambda)^2}{4\rho}\cdot 1_{n\times n})\|_F^2.\end{aligned}$$

Therefore, the optimal $\boldsymbol{K}^*$ is the projection of the matrix

$$\boldsymbol{K}_0 + \frac{1}{4\rho}(\boldsymbol{Z}^{(t)}\boldsymbol{\alpha})(\boldsymbol{Z}^{(t)}\boldsymbol{\alpha})^{\mathrm{T}} + \frac{\gamma-\lambda}{2\rho}\boldsymbol{Z}^{(t)}\boldsymbol{\alpha}\cdot 1_{n\times 1}^{\mathrm{T}} + \frac{(\gamma-\lambda)^2}{4\rho}\cdot 1_{n\times n}$$

on the cone of positive semi-definite matrix. Thus, the optimal solution to this problem is given by

$$(\boldsymbol{K}_0 + \frac{1}{4\rho}(\boldsymbol{Z}^{(t)}\boldsymbol{\alpha})(\boldsymbol{Z}^{(t)}\boldsymbol{\alpha})^{\mathrm{T}} + \frac{\gamma-\lambda}{2\rho}\boldsymbol{Z}^{(t)}\boldsymbol{\alpha}\cdot 1_{n\times 1}^{\mathrm{T}} + \frac{(\gamma-\lambda)^2}{4\rho}\cdot 1_{n\times n})_+.$$

In summary, given the labels at iteration t, we can approach the solution of Eq. (5) by optimizing Eq. (7) iteratively. Then the prediction labels at current iteration will be used as input labels in next iteration until the error rate of prediction is smaller than a pre-specified tolerance. The pseudo-code of IKMMC is shown below Algorithm 1:

Algorithm 1 Two-class IKMMC

Input: Indefinite original kernel $\boldsymbol{K}_0$, regularization parameters C, ρ and imbalance parameter l.

Output: $\boldsymbol{w}$, b, $\boldsymbol{K}$ and y

1. **Initialization**: $t = 0$, initialize the labeling vector $y \in \{\pm 1\}^n$ randomly.
2. **repeat**
3. **Initialization**: $i = 0$, $\boldsymbol{K\_set} = \phi$
   $(d_0, \boldsymbol{\alpha}_0, \lambda_0, \gamma_0) \leftarrow \max_{\boldsymbol{\alpha},\lambda,\gamma} S(\boldsymbol{\alpha}, \lambda, \gamma, (\boldsymbol{K}_0)_+)$
4. **repeat**
5. $i = i + 1$;
6. compute $\boldsymbol{K}^*$ from Eq. (8);
7. **if** $S(\boldsymbol{\alpha}_{i-1}, \lambda_{i-1}, \gamma_{i-1}, \boldsymbol{K}^*) \geqslant d_{i-1}$
8. **break;**
9. **else**
10. update $\boldsymbol{K\_set} \leftarrow \boldsymbol{K\_set} \cup \{\boldsymbol{K}^*\}$
11. **end if**
12. get $(d_i, \boldsymbol{\alpha}_i, \lambda_i, \gamma_i)$ by optimizing Eq. (7). go to step.5.
13. **until** *convergence*.
14. $t = t + 1$;
    compute $\boldsymbol{w}$ and b;
    compute $z_i^{(t)} = \mathrm{sign}(\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}_i) + b)$. go to step.3.
15. **until** *variation_ratio* $\leqslant \varepsilon$

3.3 Algorithm analysis

As mentioned above, Eqs. (6) and (7) have the same global optimal solution when there is no suboptimal kernel $\boldsymbol{K}^*$ found. The optimization procedure of Eq. (6) as well as Eq. (5) corresponds to Steps 3–13 in Algorithm 1 and the following theorem analyzes the convergence about these steps [20].

Theorem 2 Let $(d^*, \boldsymbol{\alpha}^*, \lambda^*, \gamma^*)$ be the optimal solution to the optimization problem in Eq. (6). For simplicity, we denote

$$\begin{aligned}\boldsymbol{K}_i^* &= \arg\min_{\boldsymbol{K}\geq 0} S(\boldsymbol{\alpha}_{i-1}, \lambda_{i-1}, \gamma_{i-1}, \boldsymbol{K}),\\ f(i) &= \max_{1\leqslant j\leqslant i} S(\boldsymbol{\alpha}_{j-1}, \lambda_{j-1}, \gamma_{j-1}, \boldsymbol{K}_j^*),\\ g(i) &= \max_{\boldsymbol{\alpha},\lambda,\gamma}\min_{\boldsymbol{K}_j\in\boldsymbol{K\_set}} S(\boldsymbol{\alpha}, \lambda, \gamma, \boldsymbol{K}_j),\end{aligned}$$

at ith iteration. Then we have the inequality:

$$f(i) \leqslant d^* \leqslant g(i).$$

In addition, $f(i)$ is monotonically increasing and $g(i)$ is monotonically decreasing.

Proof Notice that $\boldsymbol{K\_set}$ in Eq. (7) is a subset of $\boldsymbol{K} \geq 0$ in Eq. (6), thus we have

$$\min_{\boldsymbol{K}_j \in \boldsymbol{K\_set}} S(\boldsymbol{\alpha}, \lambda, \gamma, \boldsymbol{K}_j) \geqslant \min_{\boldsymbol{K} \geq 0} S(\boldsymbol{\alpha}, \lambda, \gamma, \boldsymbol{K}).$$

The inequality also holds for their corresponding pointwise maximum with respect to $(\boldsymbol{\alpha}, \lambda, \gamma)$:

$$\max_{\boldsymbol{\alpha},\lambda,\gamma} \min_{\boldsymbol{K}_j \in \boldsymbol{K\_set}} S(\boldsymbol{\alpha}, \lambda, \gamma, \boldsymbol{K}_j) \geqslant \max_{\boldsymbol{\alpha},\lambda,\gamma} \min_{\boldsymbol{K} \geq 0} S(\boldsymbol{\alpha}, \lambda, \gamma, \boldsymbol{K}).$$

As a result, $g(i) \geqslant d^*$ holds. Moreover, with the size of ***K_set*** increasing, $g(i)$ is monotonically decreasing.

For any feasible $(\boldsymbol{\alpha}, \lambda, \gamma)$, we denote the set of optimal values satisfying $\min_{\boldsymbol{K} \geq 0} S(\boldsymbol{\alpha}, \lambda, \gamma, \boldsymbol{K})$ as $\boldsymbol{\Omega}$. As mentioned above, d^* is the saddle point value which is the maximum value of $\min_{\boldsymbol{K} \geq 0} S(\boldsymbol{\alpha}, \lambda, \gamma, \boldsymbol{K})$ with respect to $(\boldsymbol{\alpha}, \lambda, \gamma)$, therefore d^* is the maximum value of $\boldsymbol{\Omega}$.

Since $\boldsymbol{K}_i^* = \arg\min_{\boldsymbol{K} \geq 0} S(\boldsymbol{\alpha}_{i-1}, \lambda_{i-1}, \gamma_{i-1}, \boldsymbol{K})$ holds, we have the following equality:

$$S(\boldsymbol{\alpha}_{i-1}, \lambda_{i-1}, \gamma_{i-1}, \boldsymbol{K}_i^*) = \min_{\boldsymbol{K} \geq 0} S(\boldsymbol{\alpha}_{i-1}, \lambda_{i-1}, \gamma_{i-1}, \boldsymbol{K}).$$

which means $S(\boldsymbol{\alpha}_{i-1}, \lambda_{i-1}, \gamma_{i-1}, \boldsymbol{K}_i^*)$ belongs to $\boldsymbol{\Omega}$ at the ith iteration. As a result, the following equality holds:

$$f(i) = \max_{1 \leqslant j \leqslant i} S(\boldsymbol{\alpha}_{j-1}, \lambda_{j-1}, \gamma_{j-1}, \boldsymbol{K}_j^*) \leqslant d^*.$$

Obviously, $f(i)$ increases with the addition of $\boldsymbol{K}^*$ according to the definition of $f(i)$.

For Steps 3–13, it has $f(i) \leqslant d^* \leqslant g(i)$ according to Theorem 2. As a result, we can use the gap between $f(i)$ and $g(i)$ to trace the convergence of Steps 3–13 of the algorithm. When the gap is smaller than a pre-specified value, the algorithm goes to Step 14.

Moreover, for the outer loop, the difference in error rates from two successive iterations is used as its termination criterion until the error rates is less than ε (which is set to 0.02 in the experiments). In our experiment, the outer loop can reach a stable point within ten steps. Note that the computation of the most violated $\boldsymbol{K}^*$ takes $O(n^3)$ time and the optimization problem in Eq. (7) which can be regarded as an QP problem has a time complexity of $O(n^2)$ in each iteration. So, Algorithm 1 has a time complexity of $O(n^3)$ in total.

4 Multi-class indefinite kernel maximum margin clustering

In this section, we will extend IKMMC to more sophisticated multi-class scenarios.

4.1 Model construction

Specifically, when the data set $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \in \boldsymbol{R}^d$ and the class number m are given, we can define a weight vector $\boldsymbol{w}^p$ for each class $p \in \{1, \ldots, m\}$. The multi-class MMC can therefore be formulated as [31]:

$$\begin{aligned}
&\min_{w,\xi,K} \frac{1}{2}\sum_{p=1}^{m}\|w^p\|^2 + C\sum_{i=1}^{n}\xi_i,\\
s.t.\ \ &(\sum_{p=1}^{m} z_{ip}w^p - w^r)^{\mathrm{T}}\phi(x_i) + z_{ir} \geqslant 1 - \xi_i,\\
&\forall i \in \{1, \ldots, n\}, r \in \{1, \ldots, m\},\\
&\xi_i \geqslant 0, \quad \forall i \in \{1, \ldots, n\},\\
&-l \leqslant \sum_{i=1}^{n}(w^p - w^q)^{\mathrm{T}}\phi(x_i) \leqslant l,\\
&\forall p, q \in \{1, \ldots, m\},
\end{aligned} \tag{9}$$

where the superscript p denotes the pth class and z_{ip} is defined as :

$$z_{ip} = \prod_{q=1, q\neq p}^{m} I_{[w^{pT}\phi(x_i) > w^{qT}\phi(x_i)]}, \quad \forall i \in \{1, \ldots, n\}, p \in \{1, \ldots, m\},$$

with $I(\cdot)$ denoting the indicator function. Similarly to two-class clustering, class balance constraints are added in the formula to control class imbalance.

Instead of a single PD kernel, we consider a non-PD kernel situation in this section. Similarly, we embed a F-norm regularizer measuring the diversity of the original non-PD kernel and the proxy kernel into the multi-class MMC model. The multi-class indefinite kernel maximum margin clustering can therefore be formulated as:

$$\begin{aligned}
&\min_{\boldsymbol{w},\xi,\boldsymbol{K}} \frac{1}{2}\sum_{p=1}^{m}\|\boldsymbol{w}^p\|^2 + C\sum_{i=1}^{n}\xi_i + \rho\|\boldsymbol{K} - \boldsymbol{K}_0\|_F^2,\\
s.t.\ \ &(\sum_{p=1}^{m} z_{ip}\boldsymbol{w}^p - \boldsymbol{w}^r)^{\mathrm{T}}\phi(\boldsymbol{x}_i) + z_{ir} \geqslant 1 - \xi_i,\\
&\forall i \in \{1, \ldots, n\}, r \in \{1, \ldots, m\},\\
&\xi_i \geqslant 0, \quad \forall i \in \{1, \ldots, n\},\\
&-l \leqslant \sum_{i=1}^{n}(\boldsymbol{w}^p - \boldsymbol{w}^q)^{\mathrm{T}}\phi(\boldsymbol{x}_i) \leqslant l, \quad \forall p, q \in \{1, \ldots, m\},\\
&\boldsymbol{K} \succeq 0
\end{aligned} \tag{10}$$

where

$$z_{ip} = \prod_{q=1, q\neq p}^{m} I_{[\boldsymbol{w}^{pT}\phi(\boldsymbol{x}_i) > \boldsymbol{w}^{qT}\phi(\boldsymbol{x}_i)]}, \quad \forall i \in \{1, \ldots, n\}, p \in \{1, \ldots, m\},$$

and the label for sample $\boldsymbol{x}_i$ is determined as :

$$y_i = \arg\max_p \boldsymbol{w}^{pT}\phi(\boldsymbol{x}_i) = \sum_{p=1}^{m} p z_{ip}.$$

Obviously, the objective function in Eq. (10) is also non-convex due to the first n constraints and difficult to optimize.

However, similar to two-class IKMMC, we can reach a stable point by optimizing Eq. (10) alternatively.

4.2 Optimization algorithm

Unlike two-class IKMMC, we initialize the label vector $y \in \{1,\dots,m\}^n$ by k-means algorithm. At each iteration, given initialized label vector, multi-class IKMMC is firstly transformed to a multi-class IKSVM problem with a set of class balance constraints. We also choose the semi-infinite programming strategy to optimize the IKSVM problem. Then the output label vector $y_i = \arg\max_p \boldsymbol{w}^{pT}\phi(\boldsymbol{x}_i)$ is treated as the input label at next iteration until the error ratio between two successive iterations is less than a pre-specified constant. Concretely, at iteration $t+1$, we have following objective function:

$$
\begin{aligned}
&\min_{\boldsymbol{w},\xi,\boldsymbol{K}} \frac{1}{2}\sum_{p=1}^{m}\|\boldsymbol{w}^p\|^2 + C\sum_{i=1}^{n}\xi_i + \rho\|\boldsymbol{K}-\boldsymbol{K}_0\|_F^2,\\
s.t.\quad &(\sum_{p=1}^{m} z_{ip}^{(t)}\boldsymbol{w}^p - \boldsymbol{w}^r)^{\mathrm{T}}\phi(x_i) + z_{ir}^{(t)} \geqslant 1-\xi_i,\\
&\forall i\in\{1,\dots,n\}, r\in\{1,\dots,m\},\\
&\xi_i \geqslant 0,\ \ \forall i\in\{1,\dots,n\},\\
&-l \leqslant \sum_{i=1}^{n}(\boldsymbol{w}^p-\boldsymbol{w}^q)^{\mathrm{T}}\phi(\boldsymbol{x}_i)\leqslant l,\quad \forall p,q\in\{1,\dots,m\},\\
&\boldsymbol{K}\succeq 0,
\end{aligned}
\tag{11}
$$

where $\boldsymbol{z}^{(t)} \in \boldsymbol{R}^{n\times m}$ is the output label matrix at iteration t. As two-class IKMMC, we solve the optimization in Eq. (11) by firstly reformulating it as dual formulation with respect to $(\boldsymbol{w},\boldsymbol{\xi})$:

$$
\begin{aligned}
&\max_{\boldsymbol{\alpha},\lambda,\gamma}\min_{\boldsymbol{K}} S(\boldsymbol{\alpha},\boldsymbol{\lambda},\boldsymbol{\gamma},\boldsymbol{K}),\\
s.t.\quad &\alpha_{ir}\geqslant 0,\quad \forall i\in\{1,\dots,n\}, r\in\{1,\dots,m\},\\
&0\leqslant\sum_{r=1}^{m}\alpha_{ir}\leqslant C,\quad \forall i\in\{1,\dots,n\},\\
&\lambda_{pq}\geqslant 0,\quad \gamma_{pq}\geqslant 0,\quad \forall p,q\in\{1,\dots,m\},\\
&\boldsymbol{K}\succeq 0,
\end{aligned}
\tag{12}
$$

where

$$
\begin{aligned}
&S(\boldsymbol{\alpha},\boldsymbol{\lambda},\boldsymbol{\gamma},\boldsymbol{K}) =\\
&-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{r=1}^{m}\sum_{s=1}^{m}\sum_{p=1}^{m}\alpha_{ir}\alpha_{js}z_{ip}^{(t)}z_{jp}^{(t)}K_{ij}\\
&-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{p=1}^{m}\alpha_{ip}\alpha_{jp}K_{ij}\\
&-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{r=1}^{m}\sum_{s=1}^{m}\sum_{p=1}^{m}(\lambda_{pr}-\gamma_{pr})(\lambda_{ps}-\gamma_{ps})K_{ij}\\
&-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{r=1}^{m}\sum_{s=1}^{m}\sum_{p=1}^{m}(\lambda_{rp}-\gamma_{rp})(\lambda_{sp}-\gamma_{sp})K_{ij}\\
&+\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{p=1}^{m}\sum_{q=1}^{m}\sum_{r=1}^{m}(\gamma_{pq}-\lambda_{pq})\alpha_{jr}z_{jp}^{(t)}K_{ij}\\
&+\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{r=1}^{m}\sum_{s=1}^{m}\sum_{p=1}^{m}(\lambda_{pq}-\gamma_{pq})\alpha_{jr}z_{jp}^{(t)}K_{ij}\\
&+\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{r=1}^{m}\sum_{s=1}^{m}\sum_{p=1}^{m}(\lambda_{pq}-\gamma_{pq})(\gamma_{qr}-\lambda_{qr})K_{ij}\\
&+\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{r=1}^{m}\sum_{s=1}^{m}\alpha_{ir}\alpha_{js}z_{jr}^{(t)}K_{ij}\\
&-\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{p=1}^{m}\sum_{q=1}^{m}(\gamma_{pq}-\lambda_{pq})\alpha_{jp}K_{ij}\\
&-\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{p=1}^{m}\sum_{q=1}^{m}(\lambda_{pq}-\gamma_{pq})\alpha_{jp}K_{ij}\\
&+\sum_{i=1}^{n}\sum_{r=1}^{m}\alpha_{ir}-\sum_{i=1}^{n}\sum_{r=1}^{m}\alpha_{ir}z_{ir}^{(t)}\\
&-l\sum_{p=1}^{m}\sum_{q=1}^{m}(\lambda_{pq}+\gamma_{pq})+\rho\|\boldsymbol{K}-\boldsymbol{K}_0\|_F^2,
\end{aligned}
$$

where $\boldsymbol{\alpha}\in\boldsymbol{R}^{n\times m}$, $\boldsymbol{\lambda}\in\boldsymbol{R}^{m\times m}$ and $\boldsymbol{\gamma}\in\boldsymbol{R}^{m\times m}$ are the matrixes of Lagrange dual variables.

Obviously, the optimal solution to the max-min problem in Eq. (12) is a saddle point for the function $S(\boldsymbol{\alpha},\boldsymbol{\lambda},\boldsymbol{\gamma},\boldsymbol{K})$ subject to the constraints in Eq. (12). By adding an additional variable $d\in\boldsymbol{R}$, the max-min optimization problem can be reformulated into a quadratically constrained linear program with h quadratic constraints:

$$
\begin{aligned}
&\max_{\alpha,\lambda,\gamma,d} d\\
s.t.\quad &\alpha_{ir}\geqslant 0,\quad \forall i\in\{1,\dots,n\}, r\in\{1,\dots,m\},\\
&0\leqslant\sum_{r=1}^{m}\alpha_{ir}\leqslant C,\quad \forall i\in\{1,\dots,n\},\\
&\lambda_{pq}\geqslant 0,\gamma_{pq}\geqslant 0,\forall p,q\in\{1,\dots,m\},\\
&d\leqslant S(\boldsymbol{\alpha},\boldsymbol{\lambda},\boldsymbol{\gamma},\boldsymbol{K}_j), j=1,\dots,h.
\end{aligned}
\tag{13}
$$

To approach the optimum of Eq. (12), the constraint subset will be updated based on the obtained sub-optimum in every iterative manner. At each iteration, a new PD kernel will be added into the last constraint subset in Eq. (13). With the number of constraints increasing, the solution of Eq. (13) can be infinitely close to that of Eq. (12). For notational simplicity, we have the following variable substitutions:

$$
\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{r=1}^{m}\sum_{s=1}^{m}\sum_{p=1}^{m}\alpha_{ir}\alpha_{js}z_{ip}^{(t)}z_{jp}^{(t)}K_{ij}
$$

$$
\begin{aligned}
&= Tr(((\boldsymbol{z}^{(t)}\boldsymbol{z}^{(t)\mathrm{T}}).*(\boldsymbol{\alpha}\boldsymbol{1}_{m\times m}\boldsymbol{\alpha}^{\mathrm{T}}))\boldsymbol{K}^{\mathrm{T}}) = Tr(\boldsymbol{A}\cdot\boldsymbol{K}^{\mathrm{T}}),\\
&\quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{p=1}^{m}\alpha_{ip}\alpha_{jp}K_{ij}\\
&\quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{r=1}^{m}\sum_{s=1}^{m}\sum_{p=1}^{m}(\lambda_{pr}-\gamma_{pr})(\lambda_{ps}-\gamma_{ps})K_{ij}\\
&= \boldsymbol{e}_m^{\mathrm{T}}(\lambda-\gamma)(\lambda-\gamma)\boldsymbol{e}_m Tr(\boldsymbol{K}\boldsymbol{1}_{n\times n}) = \boldsymbol{B}\cdot Tr(\boldsymbol{K}\boldsymbol{1}_{n\times n})\\
&\quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{r=1}^{m}\sum_{s=1}^{m}\sum_{p=1}^{m}(\lambda_{rp}-\gamma_{rp})(\lambda_{sp}-\gamma_{sp})K_{ij}\\
&= \boldsymbol{e}_m^{\mathrm{T}}(\lambda-\gamma)(\lambda-\gamma)^{\mathrm{T}}\boldsymbol{e}_m Tr(\boldsymbol{K}\boldsymbol{1}_{n\times n}) = \boldsymbol{C}\cdot Tr(\boldsymbol{K}\boldsymbol{1}_{n\times n})\\
&\quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{r=1}^{m}\sum_{s=1}^{m}\alpha_{ir}\alpha_{js}z_{jr}^{(t)}K_{ij},\\
&= Tr(((\boldsymbol{\alpha}\boldsymbol{z}^{(t)\mathrm{T}}).*(\boldsymbol{1}_{m\times m}\boldsymbol{\alpha}^{\mathrm{T}}))\boldsymbol{K}^{\mathrm{T}}) = Tr(\boldsymbol{D}\cdot\boldsymbol{K}^{\mathrm{T}})\\
&\quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{p=1}^{m}\sum_{q=1}^{m}\sum_{r=1}^{m}(\gamma_{pq}-\lambda_{pq})\alpha_{jr}z_{jp}^{(t)}K_{ij}\\
&= \boldsymbol{e}_m^{\mathrm{T}}\boldsymbol{K}((\boldsymbol{z}^{(t)}(\gamma-\lambda)\boldsymbol{e}_m).*(\boldsymbol{\alpha}\boldsymbol{e}_m)) = \boldsymbol{e}_m^{\mathrm{T}}\boldsymbol{K}\cdot\boldsymbol{E},\\
&\quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{p=1}^{m}\sum_{q=1}^{m}(\gamma_{pq}-\lambda_{pq})\alpha_{jp}K_{ij}\\
&= \boldsymbol{e}_m^{\mathrm{T}}\boldsymbol{K}(\gamma-\lambda)\boldsymbol{e}_m = \boldsymbol{e}_m^{\mathrm{T}}\boldsymbol{K}\cdot\boldsymbol{F},\\
&\quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{p=1}^{m}\sum_{q=1}^{m}\sum_{r=1}^{m}(\lambda_{pq}-\gamma_{pq})\alpha_{jr}z_{jq}^{(t)}K_{ij}\\
&= \boldsymbol{e}_m^{\mathrm{T}}\boldsymbol{K}((\boldsymbol{z}^{(t)}(\gamma-\lambda)^{\mathrm{T}}\boldsymbol{e}_m).*(\boldsymbol{\alpha}\boldsymbol{e}_m)) = \boldsymbol{e}_n^{\mathrm{T}}\boldsymbol{K}\cdot\boldsymbol{G},\\
&\quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{p=1}^{m}\sum_{q=1}^{m}(\lambda_{pq}-\gamma_{pq})\alpha_{jp}K_{ij}\\
&= \boldsymbol{e}_m^{\mathrm{T}}\boldsymbol{K}\boldsymbol{\alpha}(\lambda-\gamma)^{\mathrm{T}}\boldsymbol{e}_m = \boldsymbol{e}_n^{\mathrm{T}}\boldsymbol{K}\cdot\boldsymbol{H},\\
&\quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{p=1}^{m}\sum_{q=1}^{m}\sum_{r=1}^{m}(\lambda_{pq}-\gamma_{pq})(\lambda_{qr}-\gamma_{qr})K_{ij}\\
&= \boldsymbol{e}_m^{\mathrm{T}}(\lambda-\gamma)(\lambda-\gamma)\boldsymbol{e}_m Tr(\boldsymbol{K}\boldsymbol{1}_{n\times n}) = \boldsymbol{Q}\cdot Tr(\boldsymbol{K}\boldsymbol{1}_{n\times n}),
\end{aligned}
$$

where e_m is a vector of all ones of length m. As a result, $S(\boldsymbol{\alpha},\boldsymbol{\lambda},\boldsymbol{\gamma},\boldsymbol{K})$ can be further reformulated as follows:

$$
\begin{aligned}
S(\boldsymbol{\alpha},\boldsymbol{\lambda},\boldsymbol{\gamma},\boldsymbol{K}) = Tr((&-\frac{1}{2}(\boldsymbol{A}+\boldsymbol{\alpha}\boldsymbol{\alpha}^{\mathrm{T}})+(\boldsymbol{Q}-\frac{1}{2}(\boldsymbol{B}+\boldsymbol{C}))\boldsymbol{1}_{n\times n}\\
&+\boldsymbol{D}+(\boldsymbol{E}-\boldsymbol{F}+\boldsymbol{G}-\boldsymbol{H})\boldsymbol{e}_n^{\mathrm{T}})\boldsymbol{K})+\sum_{i=1}^{n}\sum_{r=1}^{m}\alpha_{ir}\\
&-\sum_{i=1}^{n}\sum_{r=1}^{m}\alpha_{ir}z_{ir}^{(t)}-l\sum_{p=1}^{m}\sum_{q=1}^{m}(\lambda_{pq}+\gamma_{pq})\\
&+\rho\|\boldsymbol{K}-\boldsymbol{K}_0\|_F^2.
\end{aligned}
$$

To obtain the optimum of Eq. (12) from a given intermediate solution pair $(\boldsymbol{\alpha},\boldsymbol{\lambda},\boldsymbol{\gamma})$, we find the next $\boldsymbol{K}\succcurlyeq 0$ with the most violated violation which can be computed by solving the minimization problem: $\min_{\boldsymbol{K}\succcurlyeq 0} S(\boldsymbol{\alpha},\boldsymbol{\lambda},\boldsymbol{\gamma},\boldsymbol{K})$ when $(\boldsymbol{\alpha},\boldsymbol{\lambda},\boldsymbol{\gamma})$ is given.

For simplicity, we further denote

$$
-\frac{1}{2}(\boldsymbol{A}+\boldsymbol{\alpha}\boldsymbol{\alpha}^{\mathrm{T}})+(\boldsymbol{Q}-\frac{1}{2}(\boldsymbol{B}+\boldsymbol{C}))\boldsymbol{1}_{n\times n}+\boldsymbol{D}+(\boldsymbol{E}-\boldsymbol{F}+\boldsymbol{G}-\boldsymbol{H})\boldsymbol{e}_n^{\mathrm{T}}
$$

as $\boldsymbol{M}$ and the minimization problem $\min_{\boldsymbol{K}\succcurlyeq 0} S(\boldsymbol{\alpha},\boldsymbol{\lambda},\boldsymbol{\gamma},\boldsymbol{K})$ can therefore be formulated as:

$$
\begin{aligned}
&\min_{\boldsymbol{K}\succcurlyeq 0} Tr(\boldsymbol{M}\boldsymbol{K})+\sum_{i=1}^{n}\sum_{r=1}^{m}\alpha_{ir}-\sum_{i=1}^{n}\sum_{r=1}^{m}\alpha_{ir}z_{ir}^{(t)}\\
&-l\sum_{p=1}^{m}\sum_{q=1}^{m}(\lambda_{pq}+\gamma_{pq})+\rho\|\boldsymbol{K}-\boldsymbol{K}_0\|_F^2. \qquad (14)
\end{aligned}
$$

Since that $\sum_{i=1}^{n}\sum_{r=1}^{m}\alpha_{ir}-\sum_{i=1}^{n}\sum_{r=1}^{m}\alpha_{ir}z_{ir}^{(t)}-l\sum_{p=1}^{m}\sum_{q=1}^{m}(\lambda_{pq}+\gamma_{pq})$, is a constant term which can be neglected, Eq. (14) is equivalent to the following problem:

$$
\begin{aligned}
&\min_{\boldsymbol{K}\succcurlyeq 0} Tr(\boldsymbol{M}\boldsymbol{K})+\rho\|\boldsymbol{K}-\boldsymbol{K}_0\|_F^2\\
&=\min_{\boldsymbol{K}\succcurlyeq 0} Tr(\boldsymbol{M}\boldsymbol{K})+\rho Tr(\boldsymbol{K}^{\mathrm{T}}\boldsymbol{K}-2\boldsymbol{K}^{\mathrm{T}}\boldsymbol{K}_0+\boldsymbol{K}_0^{\mathrm{T}}\boldsymbol{K}_0)\\
&=\min_{\boldsymbol{K}\succcurlyeq 0}\rho Tr(\boldsymbol{K}^{\mathrm{T}}\boldsymbol{K}-2\boldsymbol{K}^{\mathrm{T}}(\boldsymbol{K}_0-\frac{\boldsymbol{M}}{2\rho})+\boldsymbol{K}_0^{\mathrm{T}}\boldsymbol{K}_0) \qquad (15)
\end{aligned}
$$

After replacing the constant term $\rho Tr(\boldsymbol{K}_0^{\mathrm{T}}\boldsymbol{K}_0)$ by $\rho Tr((\boldsymbol{K}_0-\frac{\boldsymbol{M}}{2\rho})^{\mathrm{T}}(\boldsymbol{K}_0-\frac{\boldsymbol{M}}{2\rho}))$, Eq. (15) is equivalent to:

$$
\min_{\boldsymbol{K}\succcurlyeq 0}\|\boldsymbol{K}-(\boldsymbol{K}_0-\frac{\boldsymbol{M}}{2\rho})\|_F^2, \qquad (16)
$$

thus the corresponding $\boldsymbol{K}^*$ can be computed as $(\boldsymbol{K}_0-\frac{\boldsymbol{M}}{2\rho})_+$.

The pseudo-code of multi-class IKMMC is shown as Algorithm 2. Similarly, the gap between $f(x_i)$ and $g(i)$ can be used to trace the convergence of Steps 3–13 of Algorithm 2. For the outer loop, we also check if the difference in error rates from two successive iterations is less than ε (which is set to 0.02 in the experiments). In our experiment, the outer loop can still reach a stable point within ten steps. Note that the class number is far smaller than the size of the data set, the computation of $\boldsymbol{K}^*$ still take $O(n^3)$ time. In each iteration, the optimization of Eq. (13) takes $O(n^2)$ time. As a result, the algorithm of multi-class IKMMC has a complexity of $O(n^3)$ totally.

5 Experiments

In this section, we demonstrate the clustering error and Rand Index [41] of the proposed IKMMC algorithms compared with some relative algorithms on a collection of benchmark data sets.

Algorithm 2 Multi-class IKMMC

Input: Indefinite original kernel $\boldsymbol{K}_0$, regularization parameters C, ρ and Algorithm 2

Output: $\boldsymbol{w}$, $\boldsymbol{K}$ and y

1. **Initialization**: $t = 0$, initialize the label vector by k-means.
2. **repeat**
3. **Initialization**: $i = 0$, $\boldsymbol{K\_set} = \phi$
   $(d_0, \boldsymbol{\alpha}_0, \boldsymbol{\lambda}_0, \boldsymbol{\gamma}_0) \leftarrow \max_{\boldsymbol{\alpha},\boldsymbol{\lambda},\boldsymbol{\gamma}} S(\boldsymbol{\alpha}, \boldsymbol{\lambda}, \boldsymbol{\gamma}, (\boldsymbol{K}_0)_+)$
4. **repeat**
5. $i = i + 1$;
6. compute $\boldsymbol{K}^*$ from Eq. (16);
7. **if** $S(\boldsymbol{\alpha}_{i-1}, \boldsymbol{\lambda}_{i-1}, \boldsymbol{\gamma}_{i-1}, \boldsymbol{K}^*) \geqslant d_{i-1}$
8. **break;**
9. **else**
10. update $\boldsymbol{K\_set} \leftarrow \boldsymbol{K\_set} \cup \{\boldsymbol{K}^*\}$
11. **end if**
12. get $(d_i, \boldsymbol{\alpha}_i, \boldsymbol{\lambda}_i, \boldsymbol{\gamma}_i)$ by optimizing Eq. (13). go to step 5.
13. **until** *convergence*.
14. $t = t + 1$;
    compute $\boldsymbol{w}$;
    compute $y_i = \arg\max_p \boldsymbol{w}^{p\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_i)$. go to step 3.
15. **until** *variation_ratio* $\leqslant \varepsilon$

5.1 Experimental setup

The benchmark data sets used in the experiment are shown in Tables 1 and 2. Among these data sets, DNA_large, ABE_small, SAT_small, SEG_small, DNA_small, WAV_small are generated by Duan and Keerthi from the UCI collection DNA, Letter, Satellite Image (SAT), Image Segmentation (SEG) and Waveform respectively [42]. We randomly select a subset of these data sets from each class. For DNA_large, we select two classes and randomly select 481 samples.

Table 1 Description of the two-class data sets

Dataset	Size	Dimension	Class
Pima	768	8	2
Water	116	38	2
DNA_large	481	180	2
Brecancer	277	9	2
Image	1,000	18	2
Sonar	208	60	2
Wdbc	569	30	2
FlareSolar	1,066	9	2
Diabetis	768	8	2
Thyroid	215	5	2

In our experiments, the non-PSD kernel matrix is generated according to [20]. Concretely, we first generate Gaussian kernel matrix from the data with the parameter estimated via cross validation and then apply $0.1 \times (\boldsymbol{E} + \boldsymbol{E}^{\mathrm{T}})/2$ as the perturbation where $\boldsymbol{E}$ is a matrix generated randomly with zero mean and identity covariance matrix [20]. The value of balance parameter l is set to $0.03n$ for balanced data and $0.3n$ for unbalanced data [32]. The values of the kernel parameter σ, and the parameters C and ρ are all chosen via cross validation.

Table 2 Description of the multi-class data sets

Dataset	Size	Dimension	Class
Lenses	24	4	3
Seeds	210	7	3
BalanScale	625	4	3
Iris	150	4	3
Glass	214	9	6
ABE_small	600	16	3
SAT_small	597	36	6
SEG_small	250	18	7
DNA_small	600	180	3
WAV_small	600	21	3

For comparison, we use five algorithms as baselines: Clip_MMC which generates the PSD kernel matrix by neglecting the negative eigenvalues; Flip_MMC which flips the sign of the negative eigenvalues; Shift_MMC which shifts all the eigenvalues by a positive constant; kernelized k-means (KKM) [43] and iterSVM which are based on the PD kernel without applying perturbation. To make it fair, IKMMC, Clip_MMC, Flip_MMC, Shift_MMC and iterSVM have the same initial class labels.

5.2 Results on two-class data sets

Results on the various two-class data sets are summarized in Tables 3 and 4. The last line is the average value of the results of the algorithms. As can be seen, the five MMC-based algorithms basically have lower clustering error and higher rand index value. But the clustering error and rand index of IKMMC are best in these five compared clustering algorithms. It also excels PD clustering algorithm iterSVM in most cases. These results demonstrate the effectiveness of the proposed IKMMC.

Figures 1 and 2 plot the clustering errors on Pima, Image, WDBC, Diabetis with variation values for the parameters σ and C respectively. Specially, as can be seen, all the algorithms show a trend of shock and the five MMC-based algorithms have roughly similar trend of variation. But IKMMC performs better than the other four MMC-based algorithms generally. In particular, for Image and Diabetis, IKMMC leads lower clustering errors than the other algorithms in the reasonable range of the parameter σ. As for the parameter C, IKMMC still has better results, which indicates the

Table 3 Clustering error on the various two-class data sets

Dataset	Clip_MMC	Flip_MMC	Shift_MMC	KKM	iterSVM	IKMMC
Pima	0.3854±2.26	0.3438±1.97	0.3646±1.63	0.3555±2.24	0.3815±1.53	**0.3281**±1.14
Water	0.3534±1.38	0.3621±1.78	0.3707±2.02	0.4397±1.89	0.4138±1.44	**0.3448**±1.92
DNA_large	0.1663±2.97	0.2266±2.37	0.2973±2.68	0.2287±2.55	**0.1455**±2.01	0.1601±2.01
Brecancer	**0.2491**±2.20	0.2599±2.16	**0.2491**±2.17	0.2996±2.25	0.2635±2.14	**0.2491**±2.16
Image	0.3350±1.33	0.3250±1.41	0.3250±1.27	0.3580±1.18	0.3330±1.41	**0.3020**±1.12
Sonar	0.4135±1.26	**0.3894**±1.02	0.4087±1.19	0.4471±1.16	0.4519±1.36	0.4038±1.09
Wdbc	0.0826±1.81	0.0844±1.14	0.0826±1.12	**0.0773**±1.51	0.0861±1.38	0.0808±1.05
FlareSolar	0.3893±1.15	0.3959±1.20	0.3884±1.44	0.4315±1.56	0.3846±1.13	**0.3837**±1.07
Diabetis	0.3177±1.17	0.2891±1.12	0.3008±1.14	0.3294±1.64	0.2969±1.60	**0.2656**±1.15
Thyroid	0.0977±2.02	0.0977±2.42	**0.0930**±2.09	0.2279±2.28	0.1116±2.16	**0.0930**±2.05
Average	0.2790±1.76	0.2774±1.66	0.2880±1.68	0.3195±1.83	0.2868±1.62	**0.2611**±1.48

Table 4 Rand index on the various two-class data sets

Dataset	Clip_MMC	Flip_MMC	Shift_MMC	KKM	iterSVM	IKMMC
Pima	0.5256±2.11	0.5482±2.12	0.5361±2.20	0.5435±2.09	0.5275±1.89	**0.5585**±1.98
Water	0.5390±1.08	0.5340±1.07	0.5294±1.06	0.5052±1.65	0.5106±1.05	**0.5442**±1.12
DNA_large	0.7221±2.18	0.6488±2.27	0.5813±2.36	0.6465±2.54	**0.7508**±2.17	0.7305±2.09
Brecancer	**0.6245**±2.03	0.6139±2.15	0.6139±2.10	0.6050±2.37	0.6104±2.26	**0.6245**±2.20
Image	0.5470±1.39	0.5495±1.43	0.5495±1.35	0.5382±1.64	0.5553±1.22	**0.5608**±1.26
Sonar	0.5162±2.19	**0.5222**±2.13	0.5144±2.32	0.5032±2.28	0.5022±2.07	0.5162±2.10
Wdbc	0.8482±1.87	0.8452±1.68	0.8482±1.74	**0.8571**±2.02	0.8423±1.92	0.8511±1.56
FlareSolar	0.5241±2.45	0.5212±2.34	0.5245±2.58	0.5154±2.51	0.5262±2.24	**0.5266**±2.17
Diabetis	0.5659±1.56	0.5885±1.65	0.5788±1.37	0.5567±1.71	0.5820±1.23	**0.6094**±1.34
Thyroid	0.8229±2.16	0.8229±2.31	**0.8305**±2.35	0.6464±2.48	0.8007±2.25	**0.8305**±2.30
Average	0.6236±1.90	0.6194±1.92	0.6107±1.94	0.5912±2.13	0.6208±1.83	**0.6352**±1.81

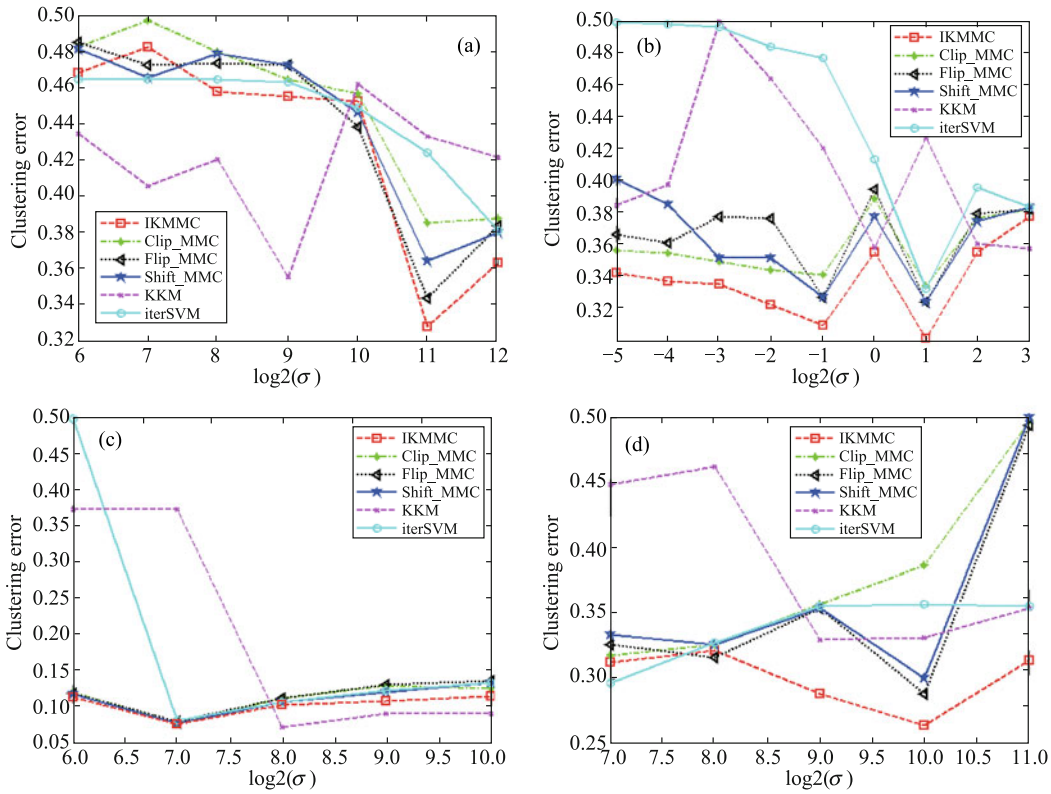


Fig. 1 Clustering error with variation values for σ on two-class data sets. (a) Pima; (b) image; (c) WDBC; (d) diabetis

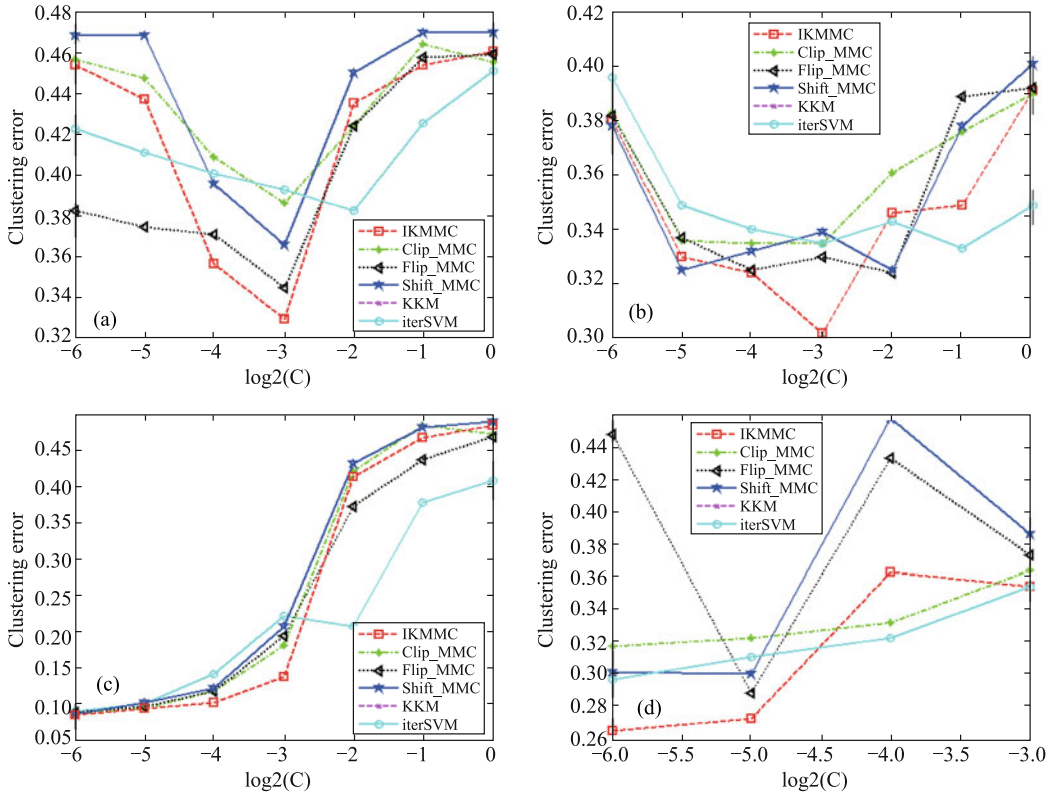


Fig. 2 Clustering error with various values for C on two-class data sets. (a) Pima; (b) image; (c) WDBC; (d) diabetis

excellent performance of IKMMC.

Figure 3 shows the convergence of two-class IKMMC on the data set Pima. The left figure plots the variation rate of prediction error of IKMMC on each iteration. The right figure plots the variation of the objective value. We can see that the prediction error rate and the objective value are both decreasing with the increasing of iteration, which verifies the effectiveness of the proposed iterative procedure. Moreover, IKMMC can reach to a stable point within 10 iterations on almost all data sets in our experiments.

5.3 Results on multi-class data sets

Results on the various multi-class data sets are summarized in Tables 5 and 6. As a whole, IKMMC still has lower clustering error and higher rand index value on multi-class data sets.

Figures 4 and 5 plot the clustering error on Lenses, BalanScale, Iris, WAV_small with variation values for the parameters σ and C respectively. As can be seen, all the algorithms show a more complex trend of shock on the multi-class data sets within the range of σ. For the four data sets, IKMMC can lead lower clustering errors than the other four MMC-based algorithms in general. Compared with KKM, IKMMC performs better on Lenses, BalanScale and Iris. We can also see that the five MMC-based algorithms are less sensitive to the parameter C overall. Moreover, in the range of C, IKMMC

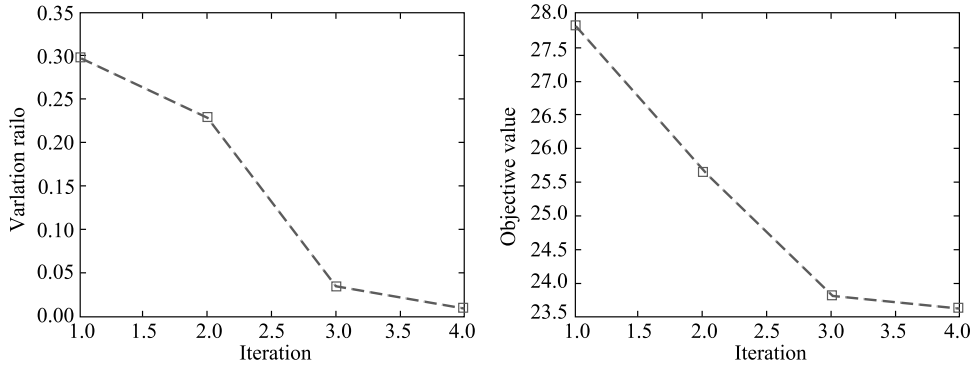


Fig. 3 Convergence of two-class IKMMC

Table 5 Clustering error on the various multi-class data sets

Dataset	Clip_MMC	Flip_MMC	Shift_MMC	KKM	iterSVM	IKMMC
Lenses	0.3750±2.66	0.3750±2.36	0.4583±2.48	0.4167±2.87	0.3750±2.09	**0.3333**±2.17
Seeds	0.1095±2.43	0.1095±2.27	**0.1048**±2.37	0.1143±2.29	0.1095±2.16	**0.1048**±2.09
BalanScale	0.3104±2.59	0.3216±2.68	0.3264±2.64	0.4208±2.93	0.3168±2.64	**0.2992**±2.52
Iris	**0.0933**±1.34	0.1000±1.23	**0.0933**±1.34	0.1000±1.23	**0.0933**±1.34	**0.0933**±1.34
Glass	0.5234±2.48	0.5187±2.52	**0.5047**±2.65	0.5187±2.71	0.5327±2.21	0.5140±2.16
ABE_small	0.4200±2.72	0.3650±2.85	0.3500±2.56	0.3467±2.81	**0.3433**±2.35	**0.3433**±2.35
SAT_small	0.3082±1.25	0.3082±1.21	0.3152±1.52	0.3099±1.79	0.3099±1.68	**0.3065**±1.28
SEG_small	0.3360±2.13	0.3280±2.24	0.3360±2.13	0.3520±2.34	0.3300±2.33	**0.3240**±2.21
DNA_small	0.2450±2.47	0.2383±2.56	0.2350±2.67	0.2383±2.85	**0.2333**±2.43	0.2367±2.47
WAV_small	0.3833±1.52	0.3867±1.65	0.4483±1.47	0.3750±1.89	0.3617±1.32	**0.3583**±1.40
Average	0.3104±2.16	0.3051±2.16	0.3172±2.18	0.3192±2.37	0.2986±2.06	**0.2913**±2.00

Table 6 Rand index on the various multi-class data sets

Dataset	Clip_MMC	Flip_MMC	Shift_MMC	KKM	iterSVM	IKMMC
Lenses	0.5471±2.86	0.5471±2.55	0.5471±2.68	0.5797±2.57	0.5471±2.16	**0.6123**±2.23
Seeds	0.8714±2.62	0.8714±2.77	**0.8762**±2.72	0.8640±2.83	0.8714±2.27	**0.8762**±2.30
BalanScale	0.6407±2.39	0.6377±2.36	0.6377±2.23	0.6341±2.41	0.6381±1.89	**0.6414**±2.07
Iris	**0.8923**±1.52	0.8859±1.64	**0.8923**±1.52	0.8859±1.64	**0.8923**±1.52	**0.8923**±1.52
Glass	0.6842±2.55	0.6793±2.34	0.6764±2.52	**0.7134**±2.49	0.6793±2.39	0.6804±2.36
ABE_small	0.6442±2.73	0.6912±2.89	0.7031±2.94	0.7066±2.98	**0.7112**±2.54	0.7104±2.46
SAT_small	0.8515±2.44	0.8517±2.50	0.8455±2.65	0.8425±2.79	0.8515±2.32	**0.8522**±2.53
SEG_small	0.8696±2.17	0.8751±2.23	0.8748±2.37	0.8737±2.45	0.8785±2.15	**0.8814**±2.12
DNA_small	0.7147±2.79	0.7246±2.85	0.7269±2.96	0.7240±3.07	**0.7282**±2.43	0.7253±2.35
WAV_small	0.6894±2.16	0.6869±2.25	0.6876±2.43	0.6814±2.59	0.6878±2.21	**0.6912**±2.18
Average	0.7405±2.42	0.7451±2.44	0.7468±2.50	0.7235±2.58	0.7485±2.19	**0.7563**±2.21

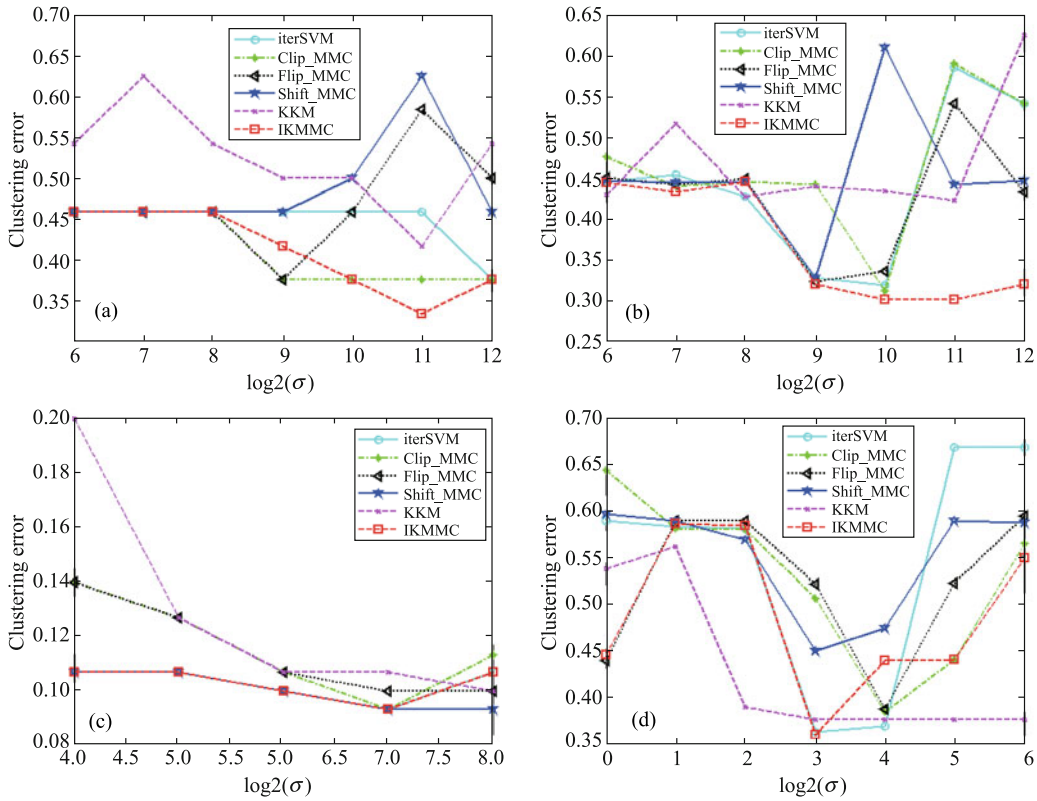


Fig. 4 Clustering error with variation values for σ on multi-class data sets. (a) Lenses; (b) BalanScale; (c) Iris; (d) WAV_small

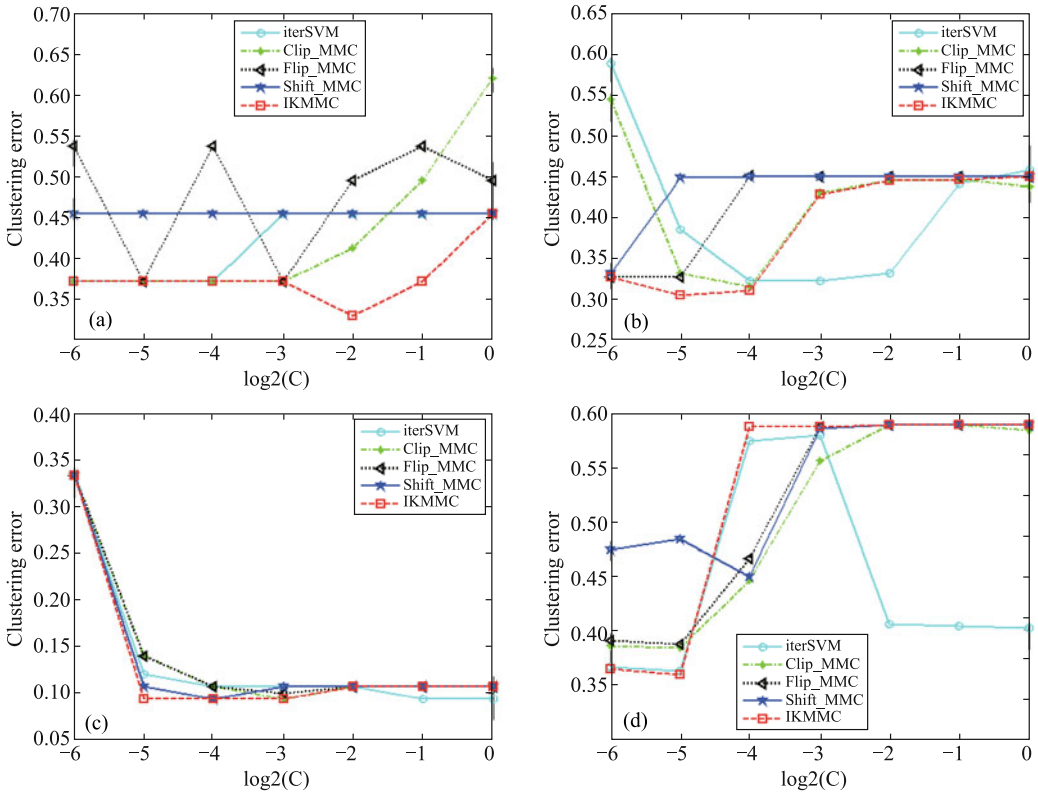


Fig. 5 Clustering error with various values for C on multi-class data sets. (a) Lenses; (b) BalanScale; (c) Iris; (d) WAV_small

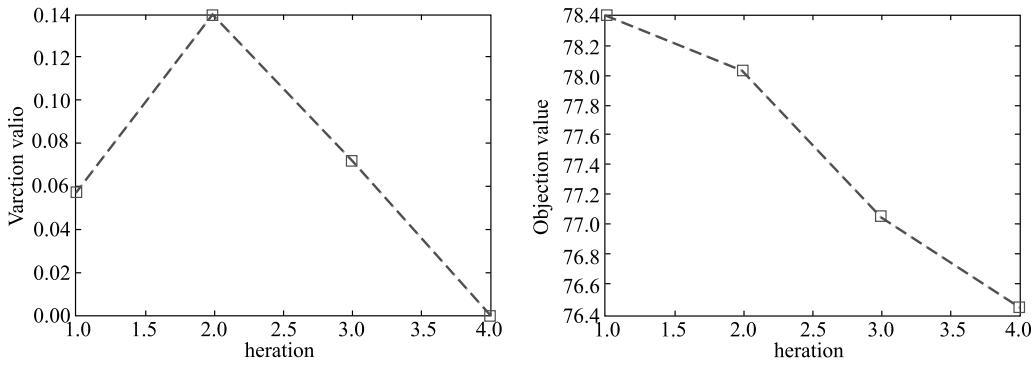


Fig. 6 Convergence of multi-class IKMMC

can still achieve a lower clustering error.

Figure 6 shows the convergence of multi-class IKMMC on BalanScale. The variation rate of prediction error of IKMMC is shown in the left figure and the right figure plots the variation of objective value with the iteration. With the increasing of iteration, the objective value is generally decreasing which demonstrates the effectiveness of the multi-class IKMMC algorithm. Moreover, IKMMC can still reach to a stable point within ten iterations on almost all multi-class data sets in our experiments.

6 Conclusion

In this paper, we focus on the indefinite kernel clustering problem due to the fact that the study of this problem is relatively scarce. The main idea of our method is to decompose the indefinite kernel clustering problem into a series of IKSVM classification problems, which actually provides some useful references for the indefinite kernel clustering.

Concretely, we propose a novel model termed as IKMMC which tries to find a proxy positive definite kernel to approxi-

mate the original indefinite one. To optimize the sophisticated non-convex objective function, we adopt an iterative strategy. Given initial labels, the indefinite kernel clustering problem can be firstly transferred to an IKSVM problem with an extra class balance constraint. The prediction labels of IKSVM are then used as the input labels at next iteration until convergence. Finally, we utilize the prediction labels at the last iteration as the expected indices of clusters. Moreover, we extend the IKMMC model from two-class scenarios to more complex multi-class scenarios. Experimental results on various data sets demonstrate the effectiveness of IKMMC.

In future, we aim to utilize multiple indefinite kernel combination instead of single kernel to further improve the performance of IKMMC. Moreover, how to develop a faster algorithm with lower complexity for IKMMC is another topic for our future research.

Acknowledgements This work was supported by the National Key R&D Program of China (2017YFB1002801), the National Natural Science Foundations of China (Grant Nos. 61375057, 61300165 and 61403193), the Natural Science Foundation of Jiangsu Province of China (BK20131298). It also supported by Collaborative Innovation Center of Wireless Communications Technology.

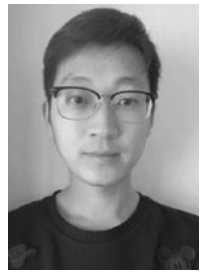
References

1. Andrew A M. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge: Cambridge University Press, 2000
2. Aronszajn N. Theory of reproducing kernels. Transactions of the American Mathematical Society, 1950, 68(3): 337–404
3. Xue H, Chen S, Yang Q. Discriminatively regularized least-squares classification. Pattern Recognition, 2009, 42(1): 93–104
4. Xue H, Chen S, Huang J. Discriminative indefinite kernel classifier from pairwise constraints and unlabeled data. In: Proceedings of International Conference on Pattern Recognition. 2012, 497–500
5. Huang J, Xue H, Zhai Y. Semi-supervised discriminatively regularized classifier with pairwise constraints. In: Proceedings of Pacific Rim International Conference on Artificial Intelligence. 2012, 112–123
6. Wang Z, Chen S, Xue H, Pan Z. A novel regularization learning for single-view patterns: multi-view discriminative regularization. Neural Processing Letters, 2010, 31(3): 159–175
7. Haasdonk B, Pekalska E. Indefinite kernel fisher discriminant. In: Proceedings of International Conference on Pattern Recognition. 2008, 1–4
8. Ho S S, Dai P, Rudzicz F. Manifold learning for multivariate variable-length sequences with an application to similarity search. IEEE Transactions on Neural Networks and Learning Systems, 2016, 27(6): 1333–1344
9. Li C, Lin L, Zuo W, Yan S, Tang J. Sold: sub-optimal low-rank decomposition for efficient video segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015, 5519–5527
10. Jacobs D W, Weinshall D, Gdalyahu Y. Classification with nonmetric distances: image retrieval and class representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(6): 583–600
11. Schleif F M, Tino P. Indefinite proximity learning: a review. Neural Computation, 2015, 27(10): 2039–2096
12. Liwicki S, Zafeiriou S, Tzimiropoulos G, Pantic M. Efficient online subspace learning with an indefinite kernel for visual tracking and recognition. IEEE Transactions on Neural Networks and Learning Systems, 2012, 23(10): 1624–1636
13. Liu C. Gabor-based kernel PCA with fractional power polynomial models for face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26(5): 572–581
14. Wu G, Chang E Y, Zhang Z. An analysis of transformation on non-positive semidefinite similarity matrix for kernel machines. In: Proceedings of the 22nd International Conference on Machine Learning. 2005, 8
15. Alabdulmohsin I, Gao X, Zhang X Z. Support vector machines with indefinite kernels. In: Proceedings of the 6th Asian Conference on Machine Learning. 2015, 32–47
16. Graepel T, Herbrich R, Bollmann-Sdorra P, Obermayer K. Classification on pairwise proximity data. In: Proceedings of the 11th Conference on Neural Information Processing Systems. 1998, 438–444
17. Roth V, Laub J, Kawanabe M, Buhmann J M. Optimal cluster preserving embedding of nonmetric proximity data. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25(12): 1540–1551
18. Luss R, d'Aspremont A. Support vector machine classification with indefinite kernels. In: Proceedings of the 20th International Conference on Neural Information Processing Systems. 2007, 953–960
19. Waldspurger I, d'Aspremont A, Mallat S. Phase recovery, maxcut and complex semidefinite programming. Mathematical Programming, 2015, 149(1–2): 47–81
20. Chen J, Ye J. Training SVM with indefinite kernels. In: Proceedings of the 25th International Conference on Machine Learning. 2008, 136–143
21. Auslender A. An exact penalty method for nonconvex problems covering, in particular, nonlinear programming, semidefinite programming, and second-order cone programming. SIAM Journal on Optimization, 2015, 25(3): 1732–1759
22. Chen Y, Gupta M R, Recht B. Learning kernels from indefinite similarities. In: Proceedings of the 26th Annual International Conference on Machine Learning. 2009, 145–152
23. Gu S, Guo Y. Learning SVM classifiers with indefinite kernels. In: Proceedings of the 26th AAAI Conference on Artificial Intelligence. 2012, 942–948
24. Lin H T, Lin C J. A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods. Neural Computation, 2003, 3: 1–32
25. Haasdonk B. Feature space interpretation of SVMs with indefinite kernels. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(4): 482–492
26. Loosli G, Ong C S, Canu S. Technical report: SVM in Krein spaces. Machine Learning, 2013
27. Ong C S. Kernels: regularization and optimization. Doctoral Thesis, The Australian National University, 2011
28. Loosli G, Canu S, Ong C S. Learning SVM in Krein spaces. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(6): 1204–1216
29. Xu H M, Xue H, Chen X, Wang Y Y. Solving indefinite kernel support

vector machine with difference of convex functions programming. In: Proceedings of the 31st AAAI Conference on Artificial Intelligence. 2017, 2782–2788

30. Xue H, Song Y, Xu H M. Multiple indefinite kernel learning for feature selection. In: Proceedings of International Joint Conferences on Artificial Intelligence. 2017, 3210–3216
31. Xu L, Neufeld J, Larson B, Schuurmans D. Maximum margin clustering. Advances in Neural Information Processing Systems, 2005, 17: 1537–1544
32. Zhang K, Tsang I W, Kwok J T. Maximum margin clustering made practical. IEEE Transactions on Neural Networks, 2009, 20(4): 583–596
33. Zhao B, Kwok J T, Zhang C. Multiple kernel clustering. In: Proceedings of the 2009 SIAM International Conference on Data Mining. 2009, 638–649
34. Wang F, Zhao B, Zhang C. Linear time maximum margin clustering. IEEE Transactions on Neural Networks, 2010, 21(2): 319–332
35. Zhang X L, Wu J. Linearithmic time sparse and convex maximum margin clustering. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2012, 42(6): 1669–1692
36. Li Y F, Tsang I W, Kwok J, Zhou Z H. Tighter and convex maximum margin clustering. In: Proceedings of International Conference on Artificial Intelligence and Statistics. 2009, 344–351
37. Wu J, Zhang X L. Sparse kernel maximum margin clustering. Neural Network World, 2011, 21(6): 551–574
38. Hettich R, Kortanek K O. Semi-infinite programming: theory, methods, and applications. SIAM Review, 1993, 35(3): 380–429
39. Smola A J, Vishwanathan S V N, Hofmann T. Kernel methods for missing variables. In: Proceedings of the 10th International Workshop on Artificial Intelligence & Statistics. 2005, 325–334
40. Joachims T, Finley T, Yu C N J. Cutting-plane training of structural SVMs. Machine Learning, 2009, 77(1): 27–59
41. Gan G, Ma C, Wu J. Data Clustering: Theory, Algorithms, and Applications. Philadelphia: SIAM, Society for Industrial and Applied Mathematics, 2007
42. Duan K B, Keerthi S S. Which is the best multiclass SVM method? An empirical study. In: Proceedings of International Workshop on Multiple Classifier Systems. 2005, 278–285
43. Filippone M, Camastra F, Masulli F, Rovetta S. A survey of kernel and spectral methods for clustering. Pattern Recognition, 2008, 41(1): 176–190

Hui Xue received the BS degree in Mathematics from Nanjing Normal University, China in 2002. She received the MS degree in Mathematics from Nanjing University of Aeronautics & Astronautics (NUAA), China in 2005. And she also received the PhD degree in Computer Application Technology at NUAA, China in 2008. Since 2009, as an associate professor, she has been with the School of Computer Science and Engineering at Southeast University, China. Her research interests include machine learning and pattern recognition.



Sen Li received the BS degree in Computer Science from Nanjing Institute of technology, China in 2012. During 2013 to 2016, he studied for a MS Degree in Computer Science at Southeast University, China. His research interests include machine learning and pattern recognition.

Xiaohong Chen received the BS degree in Mathematics from Qufu Normal University, China in 1998. In 2001, she received the MS degree in Mathematics from Nanjing University of Aeronautics & Astronautics (NUAA), China. And she also received the PhD degree in Computer Application Technology at NUAA, China in 2011. Now she is an associate professor at the College of Science at NUAA, China. Her research interests include pattern recognition and machine learning.

Yunyun Wang is an associate professor in Nanjing University of Posts and Telecommunications, China. She received her PhD in Nanjing University of Aeronautics and Astronautics, China in 2012. She joined Jiangsu Key Laboratory of Big Data Security & Intelligent Processing, China in 2017. Her current research focuses on pattern recognition and machine learning, semi-supervised learning, and transfer learning.