Multiple Kernel Clustering Based on Self-Weighted Local Kernel Alignment

Chuanli Wang[1, 2], En Zhu[1], Xinwang Liu[1], Jiaohua Qin[2], Jianping Yin[3, *] and Kaikai Zhao[4]

Abstract: Multiple kernel clustering based on local kernel alignment has achieved outstanding clustering performance by applying local kernel alignment on each sample. However, we observe that most of existing works usually assume that each local kernel alignment has the equal contribution to clustering performance, while local kernel alignment on different sample actually has different contribution to clustering performance. Therefore this assumption could have a negative effective on clustering performance. To solve this issue, we design a multiple kernel clustering algorithm based on self-weighted local kernel alignment, which can learn a proper weight to clustering performance for each local kernel alignment. Specifically, we introduce a new optimization variable- weight-to denote the contribution of each local kernel alignment to clustering performance, and then, weight, kernel combination coefficients and cluster membership are alternately optimized under kernel alignment frame. In addition, we develop a three-step alternate iterative optimization algorithm to address the resultant optimization problem. Broad experiments on five benchmark data sets have been put into effect to evaluate the clustering performance of the proposed algorithm. The experimental results distinctly demonstrate that the proposed algorithm outperforms the typical multiple kernel clustering algorithms, which illustrates the effectiveness of the proposed algorithm.

Keywords: Multiple kernel clustering, kernel alignment, local kernel alignment, self-weighted.

1 Introduction

On the one hand, kernel-based clustering algorithms are simple and effective [Filippone, Camastra, Masulli et al. (2008); Tzortzis and Likas (2009)]; on the other hand, many typical clustering algorithms, such as spectral clustering and non-negative matrix factorization clustering, can be interpreted from the perspective of kernel [Dhillon, Guan

[1] College of Computer, National University of Defense Technology, Changsha, 410073, China.

[2] College of Computer and Information Engineering, Central South University of Forestry and Technology, Changsha, 410004, China.

[3] College of Computer, Dongguan University of Technology, Dongguan, 523000, China.

[4] School of Informatics, Computing, and Engineering, Indiana University, Indiana, 47408, USA.

[*] Corresponding Author: Jianpin Yin. Email: jpyin@dgut.edu.cn.

CMC. doi:10.32604/cmc.2019.06206

and Kulis (2007); Ding, He, Simon et al. (2005)]. Therefore, kernel-based clustering algorithms have been a research hot in various applications [Gnen and Margolin (2014); Li, Qin, Xiang et al. (2015)]. Compared with one kernel, multiple kernel can provides more useful and complementary information for clustering [Cai, Nie and Huang (2013); Cai, Jiao, Zhuge et al. (2018); Hou, Nie, Tao et al. (2017)]. Multiple kernel clustering (MKC) has attracted more and more attention, and a lot of MKC algorithms and their variants have been proposed recently [Han, Yang, Yang et al. (2018); Du, Zhou, Shi et al. (2015)].

MKC algorithms aim to improve the clustering performance by jointly optimizing a group of kernel combination coefficients and cluster membership [Liu, Dou, Yin et al. (2016)]. In light of the difference of optimization frame, existing MKC algorithms can be roughly grouped into two categories. The spirit of the first category is that the single kernel is replaced with a combined kernel in objective function of clustering, and the optimal kernel combination coefficients and cluster membership are solved under clustering frame. The algorithms with regard to this one mainly include: multiple kernel *K*-means [Huang, Chuang, Chen et al. (2012)], multiple kernel fuzzy *C*-means [Chen, Chen, Lu et al. (2011)], robust multiple kernel *K*-means [Du, Zhou, Shi et al. (2015)], optimal neighborhood clustering [Liu, Zhou, Wang et al. (2017)], etc. Instead, the idea of the other category is that the cluster membership is viewed as pseudo label, and then it is put in the objective of kernel alignment [Wang, Zhao, and Tian (2015)], which is a widely used learning criterion in supervised learning, and the optimal kernel combination coefficients and pseudo label are optimized under multiple kernel learning frame. Along this idea, Lu et al. [Lu, Wang, Lu et al. (2014)] proposed centered kernel alignment for multiple kernel clustering, Liu et al. [Liu, Dou, Yin et al. (2016)] proposed kernel alignment maximization for clustering, Li et al. [Li, Liu, Wang et al. (2016)] proposed multiple kernel clustering based on local kernel alignment, etc. our work in this paper pays close attention to the clustering algorithms belonging to the second category.

Among these algorithms belonging to the second category, multiple kernel clustering based on local kernel alignment (LKAMKC) obtains prominent clustering performance by using local kernel to exploit the local structure information of data for clustering. Concretely, the sum of the objective of each local kernel alignment is defined as the optimization objective of LKAMKC, that is, it conducts local kernel alignment on each sample.

However, LKAMKC has achieved significant clustering performance, we observe that most of existing works usually assume that each local kernel alignment has an equal contribution to clustering performance, that is, each local kernel alignment is equally considered in whole clustering period. Obviously, this assumption does not well take the difference of each local kernel alignment into count, which could hinder the improving of the clustering performance. To address this issue, we propose a multiple kernel clustering algorithm based on self-weighted local kernel alignment to improve clustering performance. In detail, we introduce a weight variable to denote the contribution of each local kernel alignment to clustering performance, and then, weight of each local kernel alignment, kernel combination coefficients and cluster membership are jointly optimized. The proposed algorithm improves clustering performance by imposing learned weight on each local kernel alignment. After that, we develop a three-step alternate iterative optimization algorithm to solve the new optimization problem. Broad experiments on five benchmark data sets have been put into effect to evaluate the clustering performance of the proposed algorithm. The experimental results clearly show that the proposed algorithm outperforms the typically compared methods, which illustrates the effectiveness of the proposed algorithm.

2 Related work

In this section, we first review the related work about the kernel alignment and local kernel alignment for multiple kernel clustering.

2.1 Kernel alignment for multiple kernel clustering

Supposed a data set with n samples has m kernel feature matrices $\{\mathbf{K}_i\}_{i=1,\cdots,m}$, and the data set needs to be divided into k clusters. Let $\mathbf{H} \in \mathbb{R}^{n\times k}$ and $\mathbf{K}_\mu$ denote the relaxed cluster membership matrix and the combined kernel respectively. $\mathbf{K}_\mu$ can be calculated as:

$$\mathbf{K}_\mu = \sum_{p=1}^{m} \mu_p^2 \mathbf{K}_p \tag{1}$$

where $\mu_p \geq 0$ denotes the combination coefficient of kernel matrix $\mathbf{K}_p$ to $\mathbf{K}_\mu$.

According to Lu et al. [Lu, Wang, Lu et al. (2014)], $\mathbf{HH}^\top$ can be regarded as a pseudo ideal kernel matrix. By substituting the true deal kernel with $\mathbf{HH}^\top$, the objective of kernel alignment for multiple kernel clustering (KAMKC) can be expressed as:

$$\max_{\mathbf{H}\in\mathbb{R}^{n\times k},\mu\in\mathbb{R}_+^m} \frac{\langle \mathbf{K}_\mu, \mathbf{HH}^\top\rangle_F}{\sqrt{\langle \mathbf{K}_\mu, \mathbf{K}_\mu\rangle_F}} \ s.t. \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k,\ \boldsymbol{\mu}^\top\mathbf{1}_m = 1 \tag{2}$$

where $\langle\cdot,\cdot\rangle_F$ denotes the Frobenius inner product of the two matrices and $\boldsymbol{\mu} = [\mu_1, \cdots, \mu_m]$. $\mathbf{H}^\top\mathbf{H} = \mathbf{I}_k$ means $\mathbf{H}$ satisfies orthogonal constraint, $\boldsymbol{\mu}^\top\mathbf{1}_m = 1$ means $\boldsymbol{\mu}$ satisfies one norm constraint.

Because Eq. (2) is too complicated to directly optimize, Liu et al. [Liu, Dou, Yin et al. (2016)] not only theoretically discusses the connection between KAMKC and multiple kernel *K*-means (MKKM) but also derives an easy and equivalent optimization objective of KAMKC based on MKKM. The new optimization formula of KAMKC can be fulfilled as:

$$\begin{aligned} \min_{\mathbf{H}\in\mathbb{R}^{n\times k},\mu\in\mathbb{R}_+^m} \quad & \mathrm{Tr}\big(\mathbf{K}_\mu(\mathbf{I}_n - \mathbf{HH}^\top)\big) + \frac{\lambda}{2}\boldsymbol{\mu}^\top\mathbf{M}\boldsymbol{\mu} \\ \mathrm{s.t.} \quad & \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k,\ \boldsymbol{\mu}^\top\mathbf{1}_m = 1 \end{aligned} \tag{3}$$

where $\mathbf{M}$ with $M_{pq} = \mathrm{Tr}(\mathbf{K}_p^\top\mathbf{K}_q)$ is a positive definite matrix to measure the correlation between kernels by pairwise comparison [Cortes, Mohri and Rostamizadeh (2012)]. Regularization parameter $\lambda \geq 0$ balances the first item and the second one in objective. Clearly, if λ takes zero, KAMKC degenerates to the original MKKM.

2.2 Local kernel alignment for multiple kernel clustering

As seen from Eq. (2) or Eq. (3), KAMKC only utilizes the global structure information of kernel, while ignores its local structure information. Local kernel alignment for multiple kernel clustering (LKAMKC) enhances the clustering performance by exploiting the local structure of each sample with local kernel.

Let matrix $S^{(i)} \in \{0,1\}^{n\times\tau}$ indicates the τ nearest neighbors of i_{th} sample, and $\mathbf{K}_{\mu}^{(i)}$, $\mathbf{H}^{(i)}$ and $\mathbf{M}^{(i)}$ respectively denotes the corresponding local structure of i_{th} sample. $\mathbf{K}_{\mu}^{(i)}$, $\mathbf{H}^{(i)}$ and $\mathbf{M}^{(i)}$ can be computed respectively as follows:

$$\mathbf{K}_{\mu}^{(i)} = S^{(i)^{\top}}\mathbf{K}_{\mu}S^{(i)} \tag{4}$$

$$\mathbf{H}^{(i)} = S^{(\mathbf{i})^{\top}}\mathbf{H} \tag{5}$$

$$\mathbf{M}^{(i)}{}_{pq} = \mathrm{Tr}(\mathbf{K}_{p}{}^{(i)^{\top}}\mathbf{K}_{q}{}^{(i)}) \tag{6}$$

Replacing the global $\mathbf{K}_{\mu}$, $\mathbf{H}$ and $\mathbf{M}$ with the local $\mathbf{K}_{\mu}^{(i)}$, $\mathbf{H}^{(i)}$ and $\mathbf{M}^{(i)}$ respectively, the objective of local kernel alignment (LKA) on i_{th} sample can be written as:

$$\begin{aligned}\min_{\mathbf{H}\in\mathbb{R}^{n\times k},\boldsymbol{\mu}\in\mathbb{R}_{+}^{m}} \quad & \mathrm{Tr}\left(\mathbf{K}_{\mu}^{(i)}(\mathbf{I}_{\tau} - \mathbf{H}^{(i)}\mathbf{H}^{(i)^{\top}})\right) + \frac{\lambda}{2}\boldsymbol{\mu}^{\top}\mathbf{M}^{(i)}\boldsymbol{\mu} \\ \mathrm{s.t.} \quad & \mathbf{H}^{\top}\mathbf{H} = \mathbf{I}_{k},\ \boldsymbol{\mu}^{\top}\mathbf{1}_{m} = 1\end{aligned} \tag{7}$$

By accumulating objective of each LKA one by one, the objective of LKAMKC can be written as:

$$\begin{aligned}\min_{\mathbf{H}\in\mathbb{R}^{n\times k},\boldsymbol{\mu}\in\mathbb{R}_{+}^{m}} \quad & \sum_{i=1}^{n}\left[\mathrm{Tr}\left(\mathbf{K}_{\boldsymbol{\mu}}(A^{(i)} - A^{(i)}\mathbf{H}\mathbf{H}^{\top}A^{(i)})\right) + \frac{\lambda}{2}\boldsymbol{\mu}^{\top}M^{(i)}\boldsymbol{\mu}\right] \\ \mathrm{s.t.} \quad & \mathbf{H}^{\top}\mathbf{H} = \mathbf{I}_{k},\ \boldsymbol{\mu}^{\top}\mathbf{1}_{m} = 1\end{aligned} \tag{8}$$

where $A^{(i)} = S^{(i)}S^{(i)^{\top}}$.

3 Multiple kernel clustering based on self-weighted local kernel alignment

3.1 The proposed formulation

As shown from Eq. (8), LKAMKC equally considers each local kernel alignment, while inappropriately ignores the difference between each local kernel alignment. Thus, the contribution of each LKA to clustering performance is not properly exploited, which could hinder the improving of the clustering performance. To address the issue, we introduce a new weight variable to denote the contribution of each local kernel alignment to clustering performance. The new optimization variable and the old optimization variables in Eq. (8) are jointly optimized. By imposing a weight for each local kernel alignment on Eq. (8), the formulation of the proposed multiple kernel clustering algorithm can be written as:

$$\begin{aligned}\min_{\mathbf{H}\in\mathbb{R}^{n\times k},\boldsymbol{\mu}\in\mathbb{R}_{+}^{m},w} \quad & \sum_{i=1}^{n} w_{i}^{2}\left[\mathrm{Tr}\left(\mathbf{K}_{\mu}(A^{(i)} - A^{(i)}\mathbf{H}\mathbf{H}^{\top}A^{(i)})\right) + \frac{\lambda}{2}\mu^{\top}\mathbf{M}^{(i)}\mu\right] \\ \mathrm{s.t.} \quad & \mathbf{H}^{\top}\mathbf{H} = \mathbf{I}_{k},\ \boldsymbol{\mu}^{\top}1_{m} = 1,\ \mathbf{w}^{\top}\mathbf{1}_{n} = 1\end{aligned} \tag{9}$$

where $\mathbf{w} = [w_{1}, w_{2}, \cdots, w_{n}]$ is the weight of the each local kernel alignment, $\mathbf{w}^{\top}\mathbf{1}_{n} = 1$ means w needs satisfy one norm constraint.

3.2 Optimization

Although the proposed algorithm introduces a new variable, the optimization problem in Eq. (9) can still be solved. Specifically, we proposed a three-step alternating iterative method to optimize Eq. (9).

(i) Optimizing $\mathbf{H}$ when $\boldsymbol{\mu}$ and $\mathbf{w}$ are given

Supposed other two optimization variables are given beforehand, then Eq. (9) can be translated into the following optimization problem .

$$\max_{\mathrm{H}\in\mathbb{R}^{n\times k}} \mathrm{Tr}(\mathbf{H}^\top\mathbf{V}\mathbf{H})\ \mathrm{s.t.}\ \mathbf{H}^\top\mathbf{H}=\mathbf{I}_k \tag{10}$$

where $\mathbf{V}=\sum_{i=1}^{n} w_i{}^2\,(A^{(i)}\mathbf{K}_{\boldsymbol{\mu}}A^{(i)})$.

Eq. (10) is a standard problem of kernel k-means, and the optimal $\mathbf{H}$ can be comprised by the k eigenvectors that correspond to the k largest eigenvalues of $\mathbf{V}$.

(ii) Optimizing $\boldsymbol{\mu}$ when $\mathbf{H}$ and $\mathbf{w}$ are given

If $\mathbf{H}$ and $\mathbf{w}$ are fixed, Eq. (9) is equivalent to a quadratic programming problem about μ .

$$\min_{\boldsymbol{\mu}\in\mathbb{R}_+^m} \boldsymbol{\mu}^\top(\mathbf{2U}+\lambda\mathbf{M})\boldsymbol{\mu}\ \mathrm{s.t.}\ \boldsymbol{\mu}^\top\mathbf{1}_m=1 \tag{11}$$

where $\mathbf{U}=\mathrm{diag}([\mathrm{Tr}(\mathbf{K}_1\mathbf{D}),\cdots,\mathrm{Tr}(\mathbf{K}_m\mathbf{D})])$ with $\mathbf{D}=\sum_{i=1}^{n}(w_i{}^2(A^{(i)}-A^{(i)}\mathbf{H}\mathbf{H}^\top A^{(i)}))$ and $\mathbf{M}_{pq}=\sum_{i=1}^{n} w_i{}^2\,\mathrm{Tr}(\mathbf{K}_p A^{(i)}\mathbf{K}_q A^{(i)})$.

Eq. (11) can be effectively solved by existing off-the-shelf packages.

(iii) Optimizing $\mathbf{w}$ when $\mathbf{H}$ and $\boldsymbol{\mu}$ are given

If $\mathbf{H}$ and $\boldsymbol{\mu}$ are fixed, Eq. (9) is equivalent to the following optimization problem.

$$\min_{\mathbf{w}\in\mathbb{R}_+^n} \sum_{i=1}^{n} w_i{}^2\,a_i\ \ \mathrm{s.t.}\ \sum_{i=1}^{n} w_i=1 \tag{12}$$

where $a_i=\left[\mathrm{Tr}\big(\mathbf{K}_{\boldsymbol{\mu}}(A^{(i)}-A^{(i)}\mathbf{H}\mathbf{H}^\top A^{(i)})\big)+\frac{\lambda}{2}\boldsymbol{\mu}^\top\mathbf{M}^{(i)}\boldsymbol{\mu}\right]$.

Clearly, if a_i is greater than zero, Eq. (12) is a convex quadratic programming problem, and it has an analytic solution. By applying the KKT condition on Eq. (12), the global optimal w_i can be computed by the following.

$$w_i=\frac{\frac{1}{a_i}}{\sum_{i=1}^{n}\frac{1}{a_i}} \tag{13}$$

To prove that a_i is greater than zero, we only need to prove that $\mathrm{Tr}\big(\mathbf{K}_{\boldsymbol{\mu}}(A^{(i)}-A^{(i)}\mathbf{H}\mathbf{H}^\top A^{(i)})\big)$ is greater than zero because $\boldsymbol{\mu}^\top\mathbf{M}^{(i)}\boldsymbol{\mu}$ must be greater than zero since $\mathbf{M}^{(i)}$ is a positive definite matrix.

Theorem 1 $\mathrm{Tr}\big(\mathbf{K}_{\boldsymbol{\mu}}(\mathbf{I}_n-\mathbf{H}\mathbf{H}^\top)\big)$ is greater than zero where $\boldsymbol{\mu}$ is given [Liu, Dou, Yin et al. (2016)].

Proof: $\mathbf{H}^\top\mathbf{H}=\mathbf{I}_k$ and $\mathbf{H}\mathbf{H}^\top\mathbf{H}=\mathbf{H}$ because of $\mathbf{H}$ is an orthogonal matrix. Let h_i denote i_{th} column of matrix $\mathbf{H}$, where i >= 1 and i <= k. Clearly, $\mathbf{H}\mathbf{H}^\top h_i=h_i$ which illustrates that $\mathbf{H}\mathbf{H}^\top$ has k eigenvalue equalling to one and $n-k$ eigenvalue equaling to zero. Alike, $\mathbf{I}_n-\mathbf{H}\mathbf{H}^\top$ has $n-k$ eigenvalue equalling to one and k eigenvalue equaling to zero, so $\mathbf{I}_n-\mathbf{H}\mathbf{H}^\top$ is a positive definite matrix. In addition, $\mathbf{K}_{\boldsymbol{\mu}}$ is a positive definite kernel matrix, therefore, $\mathrm{Tr}\big(\mathbf{K}_{\boldsymbol{\mu}}(\mathbf{I}_{\mathrm{n}} - \mathbf{H}\mathbf{H}^{\top})\big)$ is greater than zero.

Theorem 2 $\mathrm{Tr}\big(\mathbf{K}_{\boldsymbol{\mu}}(\mathrm{A}^{(\mathrm{i})} - \mathrm{A}^{(\mathrm{i})}\mathbf{H}\mathbf{H}^{\top}\mathrm{A}^{(\mathrm{i})})\big)$ is greater than zero.

Proof: We have $\mathrm{A}^{(\mathrm{i})} = \mathrm{A}^{(\mathrm{i})} * \mathrm{A}^{(\mathrm{i})}$ and $\mathrm{A}^{(\mathrm{i})} = {\mathrm{A}^{(\mathrm{i})}}^{\top}$ since $\mathrm{A}^{(\mathrm{i})} = \mathrm{S}^{(\mathrm{i})}{\mathrm{S}^{(\mathrm{i})}}^{\top}$. $\mathrm{A}^{(\mathrm{i})} - \mathrm{A}^{(\mathrm{i})}\mathbf{H}\mathbf{H}^{\top}\mathrm{A}^{(\mathrm{i})} = \mathrm{A}^{(\mathrm{i})}(\mathbf{I}_{\mathrm{n}} - \mathbf{H}\mathbf{H}^{\top})\mathrm{A}^{(\mathrm{i})}$. Let y is arbitrary vector. Clearly, $\mathrm{y}^{\top}\mathrm{A}^{(\mathrm{i})}(\mathbf{I}_{\mathrm{n}} - \mathbf{H}\mathbf{H}^{\top})\mathrm{A}^{(\mathrm{i})}\mathrm{y} > 0$ because $(\mathrm{y}^{\top}\mathrm{A}^{(\mathrm{i})})^{\top} = \mathrm{A}^{(\mathrm{i})}\mathrm{y}$ and $\mathbf{I}_{\mathrm{n}} - \mathbf{H}\mathbf{H}^{\top}$ is a positive definite matrix, which is justified by theorem 1. Therefore, $\mathrm{A}^{(\mathrm{i})} - \mathrm{A}^{(\mathrm{i})}\mathbf{H}\mathbf{H}^{\top}\mathrm{A}^{(\mathrm{i})}$ is a positive definite matrix, correspondingly, $\mathrm{Tr}\big(\mathbf{K}_{\boldsymbol{\mu}}(\mathrm{A}^{(\mathrm{i})} - \mathrm{A}^{(\mathrm{i})}\mathbf{H}\mathbf{H}^{\top}\mathrm{A}^{(\mathrm{i})})\big) > 0$.

3.3 Analysis of convergence

In the proposed algorithm, the neighborhood of samples is crucial while it is difficult to exactly define during clustering. To simplify the optimization problem, we keep the neighborhood of samples fixed in the while process of optimization. By doing so, Eq. (10) is a standard kernel k-means optimization problem, Eq. (11) is a convex quadratic programming problem and Eq. (12) is also a convex quadratic programming problem. They are all convergent. Besides, the objective of the proposed algorithm has a lower bound. Therefore, the proposed clustering algorithm is convergent. The following results of experiment can illustrate the convergence of proposed algorithm.

We use Algorithm 1 to describe the implementation of the proposed algorithm, where t is the number of iteration. The input of Algorithm 1 includes kernel matrix $\{\mathbf{K}_{\mathrm{p}}\}_{\mathrm{p}=1}^{\mathrm{m}}$, the number k of clusters, regularization parameter λ and the threshold θ of convergence. The output includes the relaxed clustering membership **H**, kernel combination coefficients **μ** and the weight **w** of each local kernel alignment. The convergent condition of Algorithm 1 is that the difference of the last two objectives is less than θ.

Algorithm 1 Multiple Kernel Clustering based on Self-weighted Local Kernel Alignment

```
Input: $\{\mathbf{K}_p\}_{p=1}^{m}$, $k$, $\lambda$, and $\theta$.
Output: $\boldsymbol{\mu}$ and $\mathbf{w}$.
Initialize $\mathrm{A}^{(i)}$ for $\forall i_{th}$ samples according to one criterion of $\tau$ nearest neighbors.
Initialize $t = 1$, $\boldsymbol{\mu}^{(t)} = \mathbf{1}_m/m$ and $\mathbf{w}^{(t)} = \mathbf{1}_n/n$.
REPEAT
  Computing $\mathbf{K}_{\boldsymbol{\mu}}^{(t)}$ via Eq.(1).
  Update $\mathbf{H}^{(t)}$ via Eq.(10).
  Update $\boldsymbol{\mu}^{(t)}$ via Eq.(11).
  Update $\mathbf{w}^{(t)}$ via Eq.(13).
  $t = t + 1$.
UNTIL$(\mathrm{obj}^{(t-1)} - \mathrm{obj}^{(t)})/\mathrm{obj}^{(t)} \leq \theta$
```

4 Experiment

In this section, we conduct a large number of experiments to evaluate the clustering performance of the proposed algorithms. Moreover, we compare the proposed algorithm with many state-of-the-art MKC algorithms proposed recently.

4.1 Data sets

To conveniently and convincingly evaluate the clustering performance of the proposed algorithm, five benchmark data sets from multiple kernel learning, are adopted in our experiments. They are Yale[2], Digital[3], ProteinFold[4], Movement[5], Catech102[6]. The detailed number of samples, kernels and classes of these data sets are listed in Tab. 1.

Table 1: The details of data sets in our experiments

Dataset	#Samples	#Kernels	#Classes
Yale	165	5	15
Digital	2000	3	10
ProteinFold	694	12	27
Movement	360	12	15
Caltech102	1530	25	102

4.2 Compared algorithms

Local kernel alignment for multiple kernel clustering (LKAMKC) [Li, Liu, Wang et al. (2016)] is a strong baseline since the proposed clustering algorithm directly extends it. In addition, the compared algorithms also include many related and the state-of-the-art multiple kernel clustering algorithms. Details of compared algorithms are as follows: Multiple kernel *K*-means (MKKM) [Huang, Chuang, Chen et al. (2012)], Localized multiple kernel *K*-means (LMKKM) [Gnen and Margolin (2014)], Robust multiple kernel *K*-means (RMKKM) [Du, Zhou, Shi et al. (2015)], Co-regularized spectral clustering (CRSC) [Kumar and Daumé (2011)], Robust multi-view spectral clustering (RMSC) [Xia, Pan, Du, et al. (2014)], Robust Multiple Kernel Clustering (RMKC) [Zhou, Du, Shi et al. (2015)], Kernel alignment for multiple kernel clustering (KAMKC) [Liu, Dou, Yin et al. (2016)], Optimal kernel clustering with multiple kernels (OKMKC) [Liu, Zhou, Wang et al. (2017)].

4.3 Experiment setup

For movement data set, 12 kernel matrices are computed according to Zhou et al. [Zhou, Du, Shi et al. (2015)], and kernel matrices of the other data sets are downloaded from respective websites. To eliminate differences between kernels, we let the diagonal elements of all kernel matrices equal to one by applying centering and scaling on kernels [Cortes, Mohri and Rostamizadeh (2013)].

LKAMKC algorithm and the proposed algorithm has the same two parameters: the number of neighbors τ and regularization parameter λ. For the number of neighbors, we

[2] https://vismod.media.mit.edu/vismod/classes/mas62200/datasets/

[3] https://ss.sysu.edu.cn/ py/

[4] https://mkl.ucsd.edu/dataset/protein-fold-prediction

[5] https://archive.ics.uci.edu/ml/datasets/Libras+Movement

[6] https://mkl.ucsd.edu/dataset/

respectively select the first kernel, the second kernel, the third kernel, and the average kernel to measure the neighborhood of samples, and the optimal τ is obtained by grid search from $[0.05,0.1,\cdots,0.95]*\mathrm{n}$ where n is the number of samples. For the regularization parameter λ, the optimal value is chosen by grid search from $[2^{-10},2^{-13},\cdots,2^{10}]$. For the other compared algorithms, their parameters are set up according to the methods used in corresponding references.

To objectively evaluate the performance of the clustering algorithms, in all experiments we use the true number of classes as the number of clusters, and we adopt clustering accuracy (ACC), normalized mutual information (NMI) and purity as the indicators of the clustering performance. For all experiments, the simulations of the proposed algorithm and compared algorithms are carried out in MATLAB 2013b environment with windows 8 operation system. To reduce the effect of randomness caused by *K*-means as much as possible, we repeat each experiment for 30 times and report the best result.

Table 2: Clustering performance of all algorithms on all data sets

Datasets	MKKM	LMKKM	RMKKM	CRSC	RMSC	RMKC	KAMKC	OKMKC	LKAMKC	Proposed
					ACC					
Rale	52.12	53.33	58.79	55.14	56.35	56.97	60.00	61.21	64.24	**65.23**
Digital	48.16	48.24	46.78	73.51	81.80	80.88	83.22	83.96	91.65	**94.88**
ProteinFold	27.23	23.49	30.98	34.87	33.00	28.82	36.46	37.90	37.90	**41.43**
movement	45.12	45.03	50.79	50.05	49.17	48.10	50.00	52.78	53.24	**56.25**
Caltech102	34.77	27.97	29.67	33.33	31.50	35.56	35.82	37.32	39.48	**43.61**
					NMI					
Rale	54.16	55.59	59.70	56.89	59.11	57.69	61.29	62.27	65.10	**66.68**
Digital	48.16	48.16	46.87	73.51	81.80	80.88	83.22	83.96	91.63	**94.84**
ProteinFold	37.16	34.92	38.78	43.34	43.91	39.46	45.32	46.93	44.46	**47.68**
movement	59.17	58.59	61.70	58.89	61.11	61.69	61.29	62.27	63.10	**65.30**
Caltech102	59.64	55.17	55.86	58.20	58.40	59.90	60.38	61.41	62.67	**65.63**
					Purity					
Rale	33.86	32.71	36.60	40.78	42.36	36.46	42.65	45.24	43.95	**66.88**
Digital	49.70	49.70	44.20	77.75	82.90	88.90	90.40	91.05	96.25	**97.11**
ProteinFold	33.86	32.71	36.60	40.78	42.36	36.46	42.65	45.24	43.95	**46.76**
Movement	49.73	45.58	53.09	50.56	50.00	49.78	50.56	52.82	63.85	**56.45**
Caltech102	37.25	29.41	31.70	35.75	33.27	37.12	37.65	39.08	41.83	**45.48**

4.4 Experimental results

Tab. 2 reports the best experimental results of the proposed algorithm and all compared algorithm, and Tab. 3 reports the more detailed comparison results between the proposed algorithm and LKAMKC algorithm. In all experiment, the neighborhood of samples is fixed but the criterion to measure the neighborhood of samples is adjustable. In Tab. 2, LKAMKC and the proposed algorithm use the average combination kernel to measure the neighborhood of samples. In Tab. 3, LKAMKC-K1, LKAMKC-K2, LKAMKC-K3 and LKAMKC-A denotes LKAMKC respectively adopts the first kernel, the second kernel, the third kernel and the average combination kernel to measure the neighborhood of samples. Also proposed-K1, proposed-K2, proposed-K3 and proposed-A denotes the proposed algorithm respectively adopts the first kernel, the second kernel, the third kernel and the average combination kernel to measure the neighborhood of samples. From Table 2, we have the following observation.

Table 3: The detailed comparison between the proposed algorithm and LKAMKC

Datasets	LKAMKC-K1	LKAMKC-K2	LKAMKC-K3	LKAMKC-A	Proposed-K1	Proposed-K2	Proposed-K3	Proposed-A
				ACC				
Rale	62.42	61.11	60.00	62.50	64.24	63.03	61.82	**64.26**
Digital	89.90	86.48	92.24	93.10	91.25	89.75	95.00	**95.70**
ProteinFold	32.11	29.00	32.00	34.76	34.73	30.84	36.74	**39.48**
movement	47.20	51.00	49.10	51.05	48.17	52.24	50.09	**52.68**
Caltech102	34.55	33.97	32.46	35.33	36.74	34.77	34.44	**37.47**
				NMI				
Rale	62.16	61.59	62.10	62.89	64.30	63.13	63.84	**64.78**
Digital	48.16	48.16	46.87	73.51	84.11	82.19	88.95	**90.74**
ProteinFold	42.12	38.22	43.78	44.84	44.82	41.02	45.29	**46.58**
movement	60.00	62.59	61.20	64.11	61.45	64.58	62.29	**65.27**
Caltech102	59.65	58.17	58.86	59.20	60.24	60.62	60.22	**61.23**
				Purity				
Rale	62.86	61.72	61.60	62.98	64.85	62.42	62.42	**65.85**
Digital	89.70	86.70	92.20	92.75	91.25	89.75	95.0	**95.70**
ProteinFold	40.86	35.71	41.90	42.68	42.07	37.75	44.24	**44.81**
Movement	49.63	50.58	48.09	52.56	51.00	53.78	50.56	**55.82**
Caltech102	35.25	35.41	35.70	36.75	37.72	37.58	36.54	**38.99**

These clustering algorithms which utilize local kernel, including LKAMKC and the proposed algorithm, significantly outperform the compared ones, which do not utilize local kernel, and among them, OKMKC demonstrates the best performance. Taking the results of ACC for an example, the proposed algorithm exceeds OKMKC 4.02%, 10.92%, 3.53%, 3.47%, and 6.29% on Rale, Digital, ProteinFold, Movement and Caltech102, respectively. Similar conclusion can also be found in light of NMI and purity. It clearly indicates the importance of the local geometrical structure of data for clustering.

In terms of performance indicators: ACC, NMI and purity, the proposed algorithm obtains the best clustering performance on all data sets. Taking the results of ACC for an example, it exceeds LKAMKC, which is a strong baseline since the proposed algorithm directly extends it, by 0.99%, 3.23%, 3.53%, 3.01% and 4.13% on Rale, Digital, ProteinFold, ProteinFold, Movement and Caltech102, respectively. Also, the excellent performance of the proposed algorithm in terms of the NMI and purity can be seen from the Tab. 2, where similar observation can be found. It clearly shows the superiority of suitably utilizing the local kernel alignment.

From Tab. 3, we can draw the following points:

Both LKAMKC and the proposed algorithm are sensitive to the neighborhood of samples. Taking Digital for an example, for LKAMKC and the proposed algorithm using the third kernel to measure the neighborhood of samples can achieve the better performance than using the first kernel to measure the neighborhood of samples.

Using the average kernel to measure the neighborhood of samples can achieve the better performance than using the single kernel to measure it. Taking ACC for an example, Proposed-A and LKAMKC-A exceed Proposed-K1 and LKAMKC-K1 by 0.06% and 0.08%, 3.45% and 3.20%, 5.75% and 2.65%, 4.51% and 2.85%, 0.73% and 0.78% on Yale, Digital, ProteinFold, Movement and Caltech102, respectively, which also shows that the combined kernel can contains more information of the neighborhood of samples than the single kernel contains.

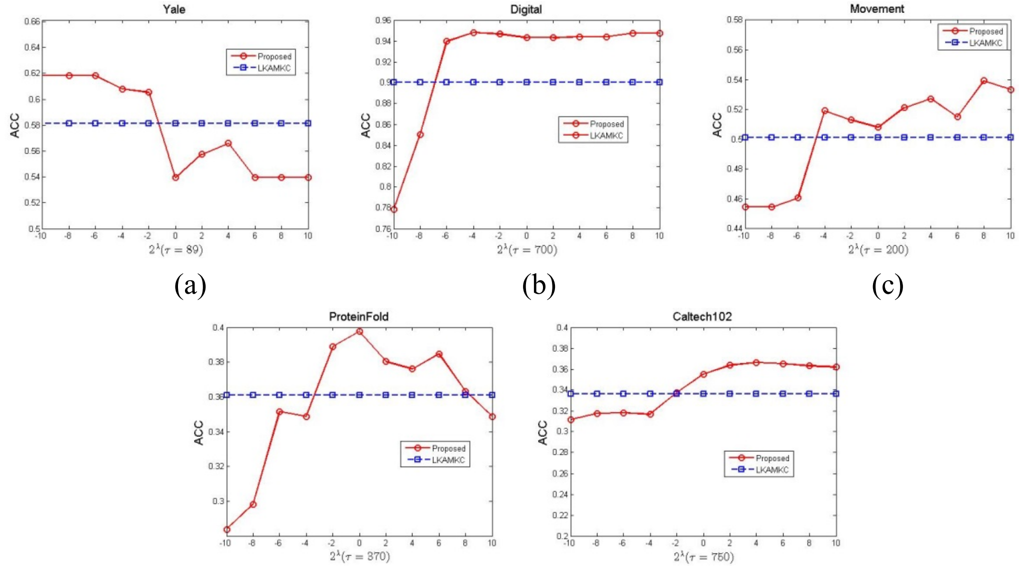
No matter which the neighborhood of samples is chosen, the proposed algorithm is always better than LKAMKC. Taking ACC for an example, Proposed-K1 exceed LKAMKC-K1 by 1.82%, 1.35%, 2.62%, 0.97%, 2.19% on Yale, Digital, ProteinFold, Movement and Caltech102, respectively, which confirms the superiority and effectiveness of the proposed algorithm again.

4.5 Parameter selection and convergence

When applying the proposed algorithm to cluster data, two parameters that contains the number τ of the nearest neighbors and regularization parameter λ need to be set up manually. Tab. 3 has analyzed the effect of the neighborhood of samples on the clustering performance. To evaluate the stability of the parameter λ, we select average kernel to measure the neighborhood of samples and fix the τ firstly and carry out a series of experiments on all data sets. Both the experimental results of the proposed algorithm and a baseline, which is the best result of LKAMKC with the same set, are drawn in Fig. 1. From Fig. 1, the following observation can be found.

1) The clustering performance of the proposed algorithm on all data sets is stable when parameter λ varies from a wide range; 2) For Yale, $\lambda = 2^{-1}$ is a watershed, if λ is less than watershed the ACC of the proposed is higher than the baseline, or the ACC of proposed is lower than the baseline. 3) For Digital and Caltech, λ also has a watershed, differently, if λ is less than watershed the ACC of proposed is lower than the baseline, or the ACC of proposed is higher than the baseline. 4) For ProteinFold and Movement, The ACC of proposed is better than the baseline when λ varies from a bounded range. For instance, when $2^{-4.5} \leq \lambda \leq 2^{8.5}$, the curve of the proposed algorithm is on the top of the baseline.

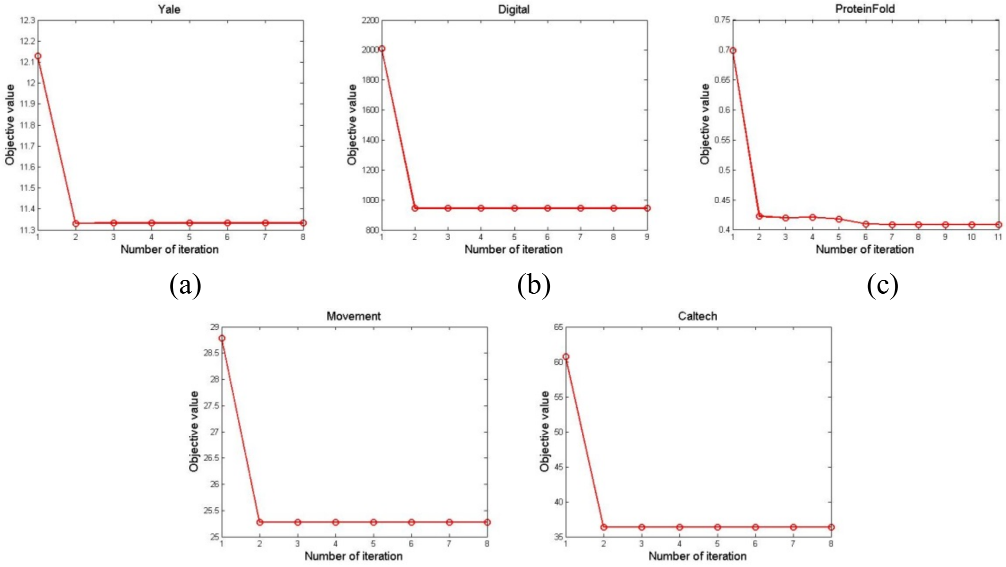
To validate the convergence of the proposed algorithm, we record the objective value of the proposed algorithms at each iteration with fixing parameter τ and λ. Fig. 2 plots the number of iteration and the corresponding objective value of the proposed algorithms at one iteration. As seen from Fig. 2, the objective value of the proposed algorithm is monotonically decreasing with regard to the time of iteration, and the proposed algorithm quickly converged in less than eleven iterations, which confirm the convergence of the proposed algorithm from the view of experiment.



(a) (b) (c)

(d) (e)

Figure 1: The performance of the proposed algorithm with regard to parameter λ



(a) (b) (c)

(d) (e)

Figure 2: The convergence of the proposed algorithm

5 Conclusions and future work

In this paper, we propose a multiple kernel clustering algorithm based on self-weighted local kernel alignment, which can improve the clustering performance by exploiting the contribution of each local kernel alignment to clustering performance more rationally. A three-step alternate optimization algorithm with convergence is developed to address the subsequent optimization problem. Broad experiments on five benchmark data sets validate the effectiveness and superiority of the proposed algorithm.

As shown from Eq. (8) and Eq. (9), both LKAMKC and the proposed algorithm utilize all local kernel to cluster. However, if the number of samples is big, the clustering algorithms based on local kernel alignment is very time-consuming. Therefore, a fast version of the proposed algorithm, which is suitable for the big data sets [Xiao, Wang, Liu et al. (2018)], is worth studying in the future.

Acknowledgement: This work was supported by the National Key R&D Program of China (No. 2018YFB1003203), National Natural Science Foundation of China (Nos. 61672528, 61773392, 61772561), Educational Commission of Hu Nan Province, China (No. 14B193) and the Key Research & Development Plan of Hunan Province (No. 2018NK2012).

References

Cai, X.; Nie, F.; Huang, H. (2013): Multi-view *K*-means clustering on big data. *International Joint Conference on Artificial Intelligence*, pp. 2598-2604.

Cai, Y.; Jiao, Y.; Zhuge, W.; Tao, H.; Hou, C. (2018): Partial multi-view spectral clustering. *Neurocomputing*, vol. 311, no. 1, pp. 316-324.

Chen, L.; Chen, C. L. P.; Lu, M. (2011): A multiple-kernel fuzzy c-means algorithm for image segmentation. *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics*, vol. 41, no. 5, pp. 1263-1274.

Cortes, C.; Mohri, M.; Rostamizadeh, A. (2012): Algorithms for learning kernels based on centered alignment. *Journal of Machine Learning Research*, vol. 13, no. 2, pp. 795-828.

Cortes, C.; Mohri, M.; Rostamizadeh, A. (2013): Multi-class classification with maximum margin multiple kernel. *Proceedings of the 30th International Conference on Machine Learning*, pp. 46-54.

Dhillon, I. S.; Guan, Y.; Kulis, B. (2007): weighted graph cuts without eigenvectors: a multilevel approach. *IEEE TPAMI*, vol. 29, no. 11, pp. 1944-1957.

Ding, C.; He, X.; Simon, H. D.; Jin, R. (2005): On the equivalence of nonnegative matrix factorization and spectral clustering. *Proceedings of the 2005 SIAM International Conference on Data Mining*, pp. 606-610.

Du, L.; Zhou, P.; Shi, L.; Wang, H., Fan, M. et al. (2015): Robust multiple kernel *K*-means using L21-norm. *International Conference on Artificial Intelligence*, pp. 3476-3482.

Filippone, M.; Camastra, F.; Masulli, F.; Rovetta, S. (2008): A survey of kernel and spectral methods for clustering. *Pattern Recognition*, vol. 41, no. 1, pp. 176-190.

Gnen, M.; Margolin, A. A. (2014): Localized data fusion for kernel *K*-means clustering with application to cancer biology. *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems*, pp. 1305-1313.

Han, Y.; Yang, K.; Yang, Y.; Ma, Y. (2018): Localized multiple kernel learning with dynamical clustering and matrix regularization. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 2, pp. 486-499.

Hou, C.; Nie, F.; Tao, H.; Yi, D. (2017): Multi-view unsupervised feature selection with adaptive similarity and view weight. *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 9, pp. 1998-2011.

Huang, H. C.; Chuang, Y. Y.; Chen, C. S. (2012): Multiple kernel fuzzy clustering. *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 1, pp. 120-134.

Kumar, A.; Iii, H. D. (2011): A co-training approach for multi-view spectral clustering. *Proceedings of the 28th International Conference on Machine Learning*, pp. 393-400.

Li, H.; Qin, J.; Xiang, X.; Pan, L.; Ma, W. et al. (2008): An efficient image matching algorithm based on adaptive threshold and RANSAC. *IEEE ACCESS*, vol. 6, pp. 66963-66971.

Li, M.; Liu, X.; Wang, L.; Dou, Y.; Yin, J. et al. (2016): Multiple kernel clustering with local kernel alignment maximization. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 1704-1710.

Liu, X.; Dou, Y.; Yin, J.; Wang, L.; Zhu, E. (2016): Multiple kernel k-means clustering with matrix-induced regularization. *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pp. 1888-1894.

Liu, X.; Zhou, S.; Wang, Y.; Li, M.; Dou, Y. et al. (2017): Optimal neighborhood kernel clustering with multiple kernels. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 2266-2272.

Lu, Y.; Wang, L.; Lu, J.; Yang, J.; Shen, C. (2014): Multiple kernel clustering based on centered kernel alignment. *Pattern Recognition*, vol. 47, no. 11, pp. 3656-3664.

Tzortzis, G. F.; Likas, A. C. (2009): The global kernel *K*-means algorithm for clustering in feature space. *IEEE Transactions on Neural Networks*, vol. 20, no. 7, pp. 1181-1194.

Wang, T.; Zhao, D.; Tian, S. (2015): An overview of kernel alignment and its applications. *Artificial Intelligence Review*, vol. 43, no. 2, pp. 179-192.

Xia, R.; Pan, Y.; Du, L.; Yin, J. (2014): Robust multi-view spectral clustering via low-rank and sparse decomposition. *Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 2149-2155.

Xiao, B.; Wang, Z.; Liu, Q.; Liu, X. (2018): SMK-means: an improved mini batch k-means algorithm based on mapreduce with big data. *Computers, Materials & Continua*, vol. 56, no. 3, pp. 365-379.

Zhou, P.; Du, L.; Shi, L.; Wang, H.; Shen, Y. D. (2015): Recovery of corrupted multiple kernels for clustering. *International Conference on Artificial Intelligence*, pp. 4105-4111.