Optimal Neighborhood Kernel Clustering with Multiple Kernels

Xinwang Liu, Sihang Zhou, Yueqing Wang, Miaomiao Li, Yong Dou, En Zhu, Jianping Yin, Han Li
School of Computer, National University of Defense Technology, Changsha, China, 410073

Abstract

Multiple kernel k-means (MKKM) aims to improve clustering performance by learning an optimal kernel, which is usually assumed to be a linear combination of a group of pre-specified base kernels. However, we observe that this assumption could: i) cause limited kernel representation capability; and ii) not sufficiently consider the negotiation between the process of learning the optimal kernel and that of clustering, leading to unsatisfying clustering performance. To address these issues, we propose an optimal neighborhood kernel clustering (ONKC) algorithm to enhance the representability of the optimal kernel and strengthen the negotiation between kernel learning and clustering. We theoretically justify this ONKC by revealing its connection with existing MKKM algorithms. Furthermore, this justification shows that existing MKKM algorithms can be viewed as a special case of our approach and indicates the extendability of the proposed ONKC for designing better clustering algorithms. An efficient algorithm with proved convergence is designed to solve the resultant optimization problem. Extensive experiments have been conducted to evaluate the clustering performance of the proposed algorithm. As demonstrated, our algorithm significantly outperforms the state-of-the-art ones in the literature, verifying the effectiveness and advantages of ONKC.

Introduction

Multiple kernel clustering (MKC) learns an optimal kernel from a group of pre-specified kernels to improve clustering performance (Zhao, Kwok, and Zhang 2009; Kumar and Daumé 2011; Xia et al. 2014; Gönen and Margolin 2014; Zhou et al. 2015; Liu et al. 2016; Li et al. 2016). These algorithms can roughly be grouped into two categories. The first category constructs a consensus matrix by utilizing low-rank optimization (Xia et al. 2014; Zhou et al. 2015; Kumar and Daumé 2011). By assuming that the optimal kernel is a linear combination of base kernels, the other category of methods optimize the combination coefficients of each base kernel by minimizing a clustering-related criterion (Yu et al. 2012; Lu et al. 2014). This category has received intensive attention during the past few years, and progress continues being made along this line of research (Gönen and Margolin 2014; Du et al. 2015; Liu et al. 2016; Li et al. 2016). In (Gönen and Margolin 2014), the kernel combination weights are allowed to adaptively change with respect to samples to better capture their individual characteristics. By replacing the squared error in k-means with an $\ell_{2,1}$-norm based one, (Du et al. 2015) presents a robust multiple kernel k-means algorithm that simultaneously finds the best clustering labels and the optimal combination of multiple kernels. The work in (Liu et al. 2016) designs a matrix-induced regularization to reduce the redundancy and enhance the diversity of the selected kernels. (Li et al. 2016) proposes a MKC algorithm with a "local" kernel alignment, which only requires that the similarity of a sample to its k-nearest neighbours be aligned with the ideal similarity matrix. Our work in this paper focuses on the second category.

One common assumption taken by the above clustering algorithms in the second category is that the optimal kernel is expressed as a linear combination of base kernels. This assumption is not only helpful to reduce the computational load of learning algorithms but also achieves promising clustering performance in partial practical applications (Yu et al. 2012; Gönen and Margolin 2014). Nevertheless, although this assumption bears the aforementioned good properties, we observe that it: i) over reduces the feasible set of optimal kernels, which could result in the learned kernel with limited representability; and ii) does not well take into account the effect of clustering on learning the optimal kernel, and ignores the possibility that these two learning processes may need to *negotiate* with each other in order to achieve the optimality. Both factors could adversely affect the learned kernel, resulting in unsatisfying clustering performance.

To address these issues, we propose an optimal neighborhood kernel clustering algorithm to enhance the representability of the learned optimal kernel. In specific, instead of rigorously requiring the optimal kernel being a linear combination of base kernels, our algorithm allows the optimal kernel to reside in the neighborhood of the latter. In this way, our algorithm effectively enlarges the region from which an optimal kernel can be chosen, and therefore is in a better position than the traditional ones to identify a more suitable kernel for clustering. Moreover, as a by-product, we theoretically show that the optimal kernel in our algorithm is dependent on both the linear combination of base kernels and the clustering result at the previous iteration. This suggests that our algorithm is able to strengthen the connection

between learning the optimal kernel and clustering automatically. The above two learning processes negotiate with each other to achieve better clustering performance. After that, we design the optimization objective of the proposed optimal neighborhood kernel clustering with multiple kernels and develop an efficient algorithm with proved convergence to solve the resultant optimization problem. Comprehensive experimental study has been conducted on 16 multiple kernel learning (MKL) benchmark data sets to compare the clustering performance of the proposed algorithm with several state-of-the-art ones. As indicated, our algorithm significantly outperforms the compared ones, validating the effectiveness and advantage of the proposed optimal neighborhood kernel clustering.

Related Work

Multiple Kernel k-means clustering (MKKM)

Let $\{\mathbf{x}_i\}_{i=1}^n \subseteq \mathcal{X}$ be a collection of n samples, and $\phi(\cdot) : \mathbf{x} \in \mathcal{X} \mapsto \mathcal{H}$ be a feature mapping which maps $\mathbf{x}$ onto a reproducing kernel Hilbert space $\mathcal{H}$. The objective of kernel k-means clustering is to minimize the sum-of-squared loss over the cluster assignment matrix $\mathbf{Z} \in \{0,1\}^{n\times k}$, which can be formulated as the following optimization problem,

$$\min_{\mathbf{Z}\in\{0,1\}^{n\times k}} \sum_{i=1,c=1}^{n,k} Z_{ic}\|\phi(\mathbf{x}_i) - \boldsymbol{\mu}_c\|_2^2 \; s.t. \; \sum_{c=1}^{k} Z_{ic} = 1, \tag{1}$$

where $n_c = \sum_{i=1}^n Z_{ic}$ and $\boldsymbol{\mu}_c = \frac{1}{n_c}\sum_{i=1}^n Z_{ic}\phi(\mathbf{x}_i)$ are the number and centroid of the c-th $(1 \le c \le k)$ cluster.

The variables $\mathbf{Z}$ in Eq.(1) is discrete, which makes the optimization problem difficult to solve. However, this problem is usually approximated through relaxing $\mathbf{Z}$ to take arbitrary real values $\mathbf{H}$, as done in the following Eq.(2),

$$\min_{\mathbf{H}\in\mathbb{R}^{n\times k}} \; \mathrm{Tr}\left(\mathbf{K}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)\right) \; s.t. \; \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \tag{2}$$

where $\mathbf{I}_k$ is an identity matrix with size $k \times k$. The optimal $\mathbf{H}$ for Eq.(2) can be obtained by taking the k eigenvectors that correspond to the k largest eigenvalues of $\mathbf{K}$.

In a multiple kernel setting, each sample has multiple feature representations via a group of feature mappings $\{\phi_p(\cdot)\}_{p=1}^m$. Specifically, each sample is represented as $\phi_{\boldsymbol{\gamma}}(\mathbf{x}) = [\sqrt{\gamma_1}\phi_1(\mathbf{x})^\top, \cdots, \sqrt{\gamma_m}\phi_m(\mathbf{x})^\top]^\top$, where $\boldsymbol{\gamma} = [\gamma_1, \cdots, \gamma_m]^\top$ denotes the coefficients of each base kernel that needs to be optimized during learning. Correspondingly, the kernel function over the above mapping function can be calculated as

$$\kappa_{\boldsymbol{\gamma}}(\mathbf{x}_i, \mathbf{x}_j) = \phi_{\boldsymbol{\gamma}}(\mathbf{x}_i)^\top\phi_{\boldsymbol{\gamma}}(\mathbf{x}_j) = \sum_{p=1}^m \gamma_p\kappa_p(\mathbf{x}_i, \mathbf{x}_j). \tag{3}$$

By replacing the kernel matrix $\mathbf{K}$ in Eq.(2) with $\mathbf{K}_{\boldsymbol{\gamma}}$ computed via Eq.(3), the following optimization objective is obtained for MKKM,

$$\begin{aligned}\min_{\mathbf{H}\in\mathbb{R}^{n\times k},\boldsymbol{\gamma}\in\mathbb{R}_+^m} \; & \mathrm{Tr}\left(\mathbf{K}_{\boldsymbol{\gamma}}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)\right) \\ s.t. \; & \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \; \sum_{p=1}^m \gamma_p^2 = 1,\end{aligned} \tag{4}$$

where an ℓ_2-norm constraint is imposed on $\boldsymbol{\gamma}$ to make this optimization bounded, and avoid only one single kernel being activated and all the others assigned with zero weights. This problem can be solved by alternately updating $\mathbf{H}$ and $\boldsymbol{\gamma}$: i) **Optimizing $\mathbf{H}$ given $\boldsymbol{\gamma}$**. With the kernel coefficients $\boldsymbol{\gamma}$ fixed, the $\mathbf{H}$ can be obtained by solving a kernel k-means clustering optimization problem in Eq.(2); ii) **Optimizing $\boldsymbol{\gamma}$ given $\mathbf{H}$**. With $\mathbf{H}$ fixed, $\boldsymbol{\gamma}$ can be analytically obtained via solving the following optimization problem

$$\min_{\boldsymbol{\gamma}\in\mathbb{R}_+^m} \; \sum_{p=1}^m \gamma_p \mathrm{Tr}\left(\mathbf{K}_p(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)\right) \; s.t. \; \sum_{p=1}^m \gamma_p^2 = 1. \tag{5}$$

MKKM with Matrix-induced Regularization (MKKM-MR)

By observing that existing MKKM algorithms do not sufficiently consider the correlation among base kernels, the work in (Liu et al. 2016) proposes to reduce the redundancy and enhance the diversity of selected kernels by incorporating a matrix-induced regularization, as fulfilled in the following Eq.(6)

$$\begin{aligned}\min_{\mathbf{H}\in\mathbb{R}^{n\times k},\boldsymbol{\gamma}\in\mathbb{R}_+^m} \; & \mathrm{Tr}\left(\mathbf{K}_{\boldsymbol{\gamma}}(\mathbf{I}_n - \mathbf{H}\mathbf{H}^\top)\right) + \frac{\lambda}{2}\boldsymbol{\gamma}^\top\mathbf{M}\boldsymbol{\gamma} \\ s.t. \; & \mathbf{H}^\top\mathbf{H} = \mathbf{I}_k, \; \boldsymbol{\gamma}^\top\mathbf{1}_m = 1,\end{aligned} \tag{6}$$

where $\mathbf{M}$ is a matrix to measure the correlation of each pairwise base kernels and λ is a parameter to trade off the clustering cost function and the regularization term.

Note that both MKKM and the newly proposed MKKM-MR algorithms take the common assumption that the optimal kernel is a linear combination of base kernels. In the following, we propose an optimal neighborhood kernel clustering with multiple kernels to improve the representation capability of the optimal kernel and enhance the negotiation between learning the optimal kernel and clustering.

The Proposed Optimal Neighborhood Kernel Clustering with Multiple Kernels

As seen from Eq.(4) and Eq.(6), the optimal kernel $\mathbf{K}_{\boldsymbol{\gamma}}$ in both MKKM and MKKM-MR is expressed as $\sum_{p=1}^m \gamma_p\mathbf{K}_p$. This assumption rigorously requires that the optimal kernel is on a hyperplane parameterized by $\boldsymbol{\gamma}$, which substantially hurts the representation capability of the optimal kernel. On the other hand, this expression, i.e., $\sum_{p=1}^m \gamma_p\mathbf{K}_p$, does not sufficiently or explicitly consider the effect of clustering matrix $\mathbf{H}$ on learning the optimal kernel. This makes the underlying connections between learning the optimal kernel and clustering loosen, and ignores the possibility that the above two learning processes may need to negotiate with each other in order to achieve the optimality.

Following the above analysis, we can see that existing MKKM algorithm and its variants do not take a sufficient consideration of the form of the optimal kernel, which could lead to unsatisfying clustering performance. This motivates us to derive an optimal neighborhood kernel clustering algorithm to improve this situation.

The Proposed Formulation

To address the above-mentioned issues, we propose to incorporate the optimal neighborhood kernel learning (Liu et al. 2009) into existing MKKM algorithms to enhance the representability of the optimal kernel and strengthen the negotiation between kernel learning and clustering. Specifically, our algorithm seeks an optimal kernel $\mathbf{G}$ in the neighbor of $\mathbf{K}_{\boldsymbol{\gamma}}$, and uses it for clustering. This idea can be fulfilled as follows

$$\begin{aligned}\min_{\mathbf{H},\boldsymbol{\gamma},\mathbf{G}} \ & \operatorname{Tr}\left(\mathbf{G}(\mathbf{I}_n-\mathbf{H}\mathbf{H}^\top)\right)+\frac{\rho}{2}\|\mathbf{G}-\mathbf{K}_{\boldsymbol{\gamma}}\|_{\mathrm{F}}^2+\frac{\lambda}{2}\boldsymbol{\gamma}^\top\mathbf{M}\boldsymbol{\gamma}\\ s.t.\ & \mathbf{H}\in\mathbb{R}^{n\times k},\ \mathbf{H}^\top\mathbf{H}=\mathbf{I}_k,\\ & \boldsymbol{\gamma}^\top\mathbf{1}_m=1,\ \gamma p\geq 0,\ \forall p,\\ & \mathbf{G}\succeq 0,\end{aligned} \tag{7}$$

where $\mathbf{G}$ is an optimal kernel which is required to be PSD, $\mathbf{K}_{\boldsymbol{\gamma}}=\sum_{p=1}^m\gamma_p\mathbf{K}_p$, the distance of $\mathbf{G}$ and $\mathbf{K}_{\boldsymbol{\gamma}}$ is measured by $\|\mathbf{G}-\mathbf{K}_{\boldsymbol{\gamma}}\|_{\mathrm{F}}^2$, $\mathbf{M}$ is a matrix with element M_{pq} measuring the correlation between $\mathbf{K}_p$ and $\mathbf{K}_q$, and $\rho,\ \lambda$ are regularization parameters.

Compared with the objective in Eq.(6), Eq.(7) has an extra variable $\mathbf{G}$ to optimize. Actually, the linear combination of base kernels $\mathbf{K}_{\boldsymbol{\gamma}}$ can be treated as a noisy observation of the ideal kernel $\mathbf{G}$, and we expect to seek a better kernel in the neighborhood of $\mathbf{K}_{\boldsymbol{\gamma}}$ for clustering. By this way, our algorithm effectively enlarges the region from which an optimal kernel can be chosen, and therefore is in a better position than the traditional ones to identify a more suitable kernel for clustering. More importantly, as will be seen in Eq.(9), the update of $\mathbf{G}$ is dependent on both the combined kernel $\mathbf{K}_{\boldsymbol{\gamma}}$ and the clustering matrix $\mathbf{H}$ at the previous iteration, which is different from existing MKKM algorithms. This indicates that in our algorithm the clustering matrix $\mathbf{H}$ is explicitly utilized to learn an optimal kernel, which, in turn, is used for clustering. These two learning processes are seamlessly coupled and are allowed to negotiate with each other to achieve better clustering.

Alternate optimization

In the following, we design an efficient algorithm to solve the optimization problem in Eq.(7). In specific, we design a three-step algorithm to solve it in an alternate manner:

i) **Optimizing $\mathbf{H}$ with fixed $\boldsymbol{\gamma}$ and $\mathbf{G}$.** Given $\boldsymbol{\gamma}$ and $\mathbf{G}$, the optimization in Eq.(7) w.r.t $\mathbf{H}$ reduces to a standard kernel k-means problem as follows

$$\min_{\mathbf{H}}\ \operatorname{Tr}\left(\mathbf{G}(\mathbf{I}_n-\mathbf{H}\mathbf{H}^\top)\right)\ \ s.t.\ \mathbf{H}\in\mathbb{R}^{n\times k},\ \mathbf{H}^\top\mathbf{H}=\mathbf{I}_k. \tag{8}$$

ii) **Optimizing $\mathbf{G}$ with fixed $\boldsymbol{\gamma}$ and $\mathbf{H}$.** Given $\boldsymbol{\gamma}$ and $\mathbf{H}$, the optimization in Eq.(7) w.r.t $\mathbf{G}$ can be rewritten as,

$$\min_{\mathbf{G}}\ \frac{1}{2}\|\mathbf{G}-\mathbf{B}\|_{\mathrm{F}}^2\ \ s.t.\ \mathbf{G}\succeq 0, \tag{9}$$

where $\mathbf{B}=\mathbf{K}_{\boldsymbol{\gamma}}-\frac{1}{\rho}(\mathbf{I}_n-\mathbf{H}\mathbf{H}^\top)$.

This optimization problem in Eq.(9) is to find the projection of $\mathbf{B}$ in PSD space. According to the Theorem 2 in (Zhou et al. 2015), its optimal solution can be readily written as $\mathbf{G}=\mathbf{U}_B\boldsymbol{\Sigma}_B^+\mathbf{V}_B^\top$, where $\mathbf{B}=\mathbf{U}_B\boldsymbol{\Sigma}_B\mathbf{V}_B^\top$ is the SVD of $\mathbf{B}$, and $\boldsymbol{\Sigma}_B^+$ is a diagonal matrix by keeping the positive elements of $\boldsymbol{\Sigma}_B$ and setting the rest ones as zeros.

iii) **Optimizing $\boldsymbol{\gamma}$ with fixed $\mathbf{H}$ and $\mathbf{G}$.** Given $\mathbf{H}$ and $\mathbf{G}$, the optimization in Eq.(7) w.r.t $\boldsymbol{\gamma}$ is a quadratic programming with linear constraints as follows,

$$\min_{\boldsymbol{\gamma}}\ \frac{\rho+\lambda}{2}\boldsymbol{\gamma}^\top\mathbf{M}\boldsymbol{\gamma}-\mathbf{a}^\top\boldsymbol{\gamma}\ \ s.t.\ \boldsymbol{\gamma}^\top\mathbf{1}_m=1,\ \gamma_p\geq 0,\ \forall p, \tag{10}$$

where $\mathbf{M}$ is a $m\times m$ matrix with $M_{pq}=\operatorname{Tr}(\mathbf{K}_p\mathbf{K}_q)$, and $\mathbf{a}=[a_1,\cdots,a_m]^\top$ with $a_p=\rho\operatorname{Tr}(\mathbf{G}\mathbf{K}_p)$.

Algorithm 1 Proposed Optimal Neighborhood Kernel Clustering with Multiple Kernels

Input: $\{\mathbf{K}_p\}_{p=1}^m$, ρ, λ and ϵ_0.
Output: $\mathbf{H}$, $\boldsymbol{\mu}$ and $\mathbf{G}$.
Initialize $\boldsymbol{\gamma}^{(0)}=\mathbf{1}_m/m$, $\mathbf{G}^{(0)}=\mathbf{K}_{\boldsymbol{\gamma}^{(0)}}$ and $t=1$.
repeat
  $\mathbf{K}_{\boldsymbol{\gamma}^{(t)}}=\sum_{p=1}^m\gamma_p^{(t-1)}\mathbf{K}_p$.
  Update $\mathbf{H}^{(t)}$ by solving Eq.(8) with given $\mathbf{G}^{(t)}$.
  Update $\mathbf{G}^{(t)}$ with $\mathbf{K}_{\boldsymbol{\gamma}^{(t)}}$ and $\mathbf{H}^{(t)}$ by Eq.(9).
  Update $\boldsymbol{\gamma}^{(t)}$ by solving Eq.(10) with given $\mathbf{H}^{(t)}$ and $\mathbf{G}^{(t)}$.
  $t=t+1$.
until $\left(\text{obj}^{(t-1)}-\text{obj}^{(t)}\right)/\text{obj}^{(t)}\leq\epsilon_0$

In sum, our algorithm for solving Eq.((7)) is outlined in Algorithm 1, where $\text{obj}^{(t)}$ denotes the objective value at the t-th iteration. It is worth pointing out that the objective of Algorithm 1 is guaranteed to be monotonically decreased when optimizing one variable with others fixed at each iteration (Bezdek and Hathaway 2003). At the same time, the objective is lower-bounded by zero. As a result, our algorithm is guaranteed to converge. Also, as shown in the experimental study, it usually converges in less than 30 iterations.

Discussion and Extension

In this subsection, we analyze the relationship between our algorithm and existing ones in the literature. The potential extension and the computational complexity of our algorithm are also discussed.

The critical difference between most of existing MKKM algorithms and ours lies at the form of the optimal kernel. In detail, existing MKKM algorithms adopt the assumption that the optimal kernel is a linear combination of base kernels. Differently, our algorithm further relaxes this assumption and only requires that the optimal kernel resides in the neighborhood of the linear combination of base kernels. As seen, existing MKKM algorithms (Huang, Chuang, and Chen 2012; Liu et al. 2016) can be treated as a special case of ours when the parameter ρ approaches to $+\infty$. In summary, our algorithm extends the existing MKKM algorithms to a more general framework, with richer kernel representability and more negotiation between the kernel learning and clustering.

Our algorithm is readily extendable by finely designing appropriate criterion to measure the neighborhood between

$\mathbf{G}$ and $\mathbf{K}_\gamma$. Designing proper criteria to satisfy various requirements of clustering tasks is interesting and worth exploring in future. In addition, the proposed ONKC is general, and can be readily extended to other kernel-based algorithms such as kernel κ-means, spectral clustering, etc.

Compared with MKKM, our algorithm needs to optimize the optimal neighborhood kernel $\mathbf{G}$ by performing SVD at each iteration, which brings a little extra computation cost. Overall, the computational complexity of existing MKKM and ours is comparable.

Experiments

Data sets

We evaluate the clustering performance of the proposed algorithm on 16 benchmark data sets from various applications, including image recognition, gesture recognition, protein subcellular localization. The detailed information of these data sets is listed in Table 1. From this tabler, we observe that the number of samples, kernels and categories of these data sets show considerable variations, which provides a good platform to compare the performance of different clustering algorithms.

Table 1: Datasets used in our experiments.

Dataset	#Samples	#Kernels	#Classes
psortPos	541	69	4
psortNeg	1444	69	5
plant	940	69	4
nonplant	2732	69	3
Digital	2000	3	10
Flower17	1360	7	17
ProteinFold	649	12	27
Flower102	8189	4	102
Caltech102	1530	25	102
warpAR10P	130	12	10
YALE	165	12	15
TOX171	171	12	4
Carcinom	174	12	11
warpPIE10P	210	12	10
JAFFE	213	12	10
movement	360	12	15

We then show how to construct base kernels for these data sets. For the first nine data sets, all kernel matrices are pre-computed and publicly available from websites[1,2,3]. For each of the rest data sets, we generate 12 base kernels by following the approach in (Du et al. 2015), including seven Gaussian kernels, four polynomial kernels and one cosine kernel.

Compared algorithms

The proposed algorithm is compared with nine multiple kernel clustering related algorithms, most of which are newly proposed. They include

- **Average multiple kernel k-means (A-MKKM)**: A new kernel is generated by uniformly weighting all base kernels, and this new kernel is taken as the input of kernel k-means.
- **Single best kernel k-means (SB-KKM)**: Kernel k-means is performed on each single kernel separately and the best result is reported.
- **Multiple kernel k-means (MKKM)** (Huang, Chuang, and Chen 2012): The algorithm performs kernel k-means and updates kernel coefficients alternately, as shown in Eq.(4).
- **Localized multiple kernel k-means (LMKKM)** (Gönen and Margolin 2014): LMMKM combines the base kernels by sample-adaptive weights.
- **Robust multiple kernel k-means (RMKKM)** (Du et al. 2015): RMKKM improves the robustness of MKKM by replacing the sum-of-squared loss with an $\ell_{2,1}$-norm one.
- **Co-regularized spectral clustering (CRSC)** (Kumar and Daumé 2011): CRSC provides a co-regularization way to perform spectral clustering.
- **Robust multiview spectral clustering (RMSC)** (Xia et al. 2014): RMSC constructs a transition probability matrix from each single view, and uses them to recover a shared low-rank transition matrix for clustering.
- **Robust Multiple Kernel Clustering (RMKC)** (Zhou et al. 2015): RMKC learns a robust yet low-rank kernel for clustering by capturing the structure of noises in multiple kernels.
- **Multiple kernel k-means with matrix-induced regularization (MKKM-MR)** (Liu et al. 2016): MKKM-MR learns the optimal combination weights by introducing a matrix-induced regularization to reduce the redundancy and enhance the diversity among the selected kernels.

The Matlab implementation of KKM, MKKM and LMKKM are publicly available from the website[4]. For RMKKM, CRSC, RMSC, RMKC and MKKM-MR, we use their Matlab codes which are freely downloaded from authors' websites in our experiments.

Experimental settings

In all our experiments, all base kernels are first centered and then scaled so that for all i and p we have $K_p(\mathbf{x}_i, \mathbf{x}_i) = 1$ by following (Cortes, Mohri, and Rostamizadeh 2012; Liu et al. 2016). For all data sets, it is assumed that the true number of clusters is known and set as the true number of classes. The parameters of RMKKM, RMSC and RMKC are selected by grid search according to the suggestions in their papers. For the proposed algorithm, its regularization parameters λ and ρ are both chosen from a large enough range $[2^{-15}, 2^{-13}, \cdots, 2^{15}]$ by grid search.

The clustering performance of all compared algorithms are evaluated in terms of three widely used criteria, including clustering accuracy (ACC), normalized mutual information (NMI) and purity. For all algorithms, we repeat each

[1] http://mkl.ucsd.edu/dataset/

[2] http://ss.sysu.edu.cn/~py/

[3] http://www.robots.ox.ac.uk/~vgg/data/flowers/

[4] https://github.com/mehmetgonen/lmkkmeans

Table 2: ACC, NMI and purity comparison of different clustering algorithms on 12 benchmark data sets.

Datasets	A-MKKM	SB-KKM	MKKM (Huang et al. 2012)	LMKKM (Gönen and Alpaydın 2014)	RMKKM (Du et al. 2015)	CRSC (Kumar and Daumé) 2011	RMSC (Xia et al. 2014)	RMKC (Zhou et al. 2015)	MKKM-MR (Liu et al. 2016)	Proposed
ACC										
Digital	88.75	75.40	47.00	47.00	40.45	84.80	90.40	88.90	90.40	**91.05**
Flower17	51.03	42.06	45.37	42.94	48.38	52.72	53.90	52.35	60.00	**60.88**
ProteinFold	28.10	33.86	27.23	23.49	30.98	34.87	33.00	28.82	36.46	**37.90**
Flower102	27.29	33.13	21.96	22.57	28.17	37.26	32.97	33.54	39.91	**41.56**
Caltech102	35.56	33.14	34.77	27.97	29.67	33.33	31.50	35.56	35.82	**37.32**
warpAR10P	39.23	43.08	41.54	27.69	31.54	33.08	30.77	39.23	40.77	**47.69**
YALE	52.12	56.97	52.12	53.33	58.79	55.15	56.36	56.97	60.00	**61.21**
TOX171	47.95	47.95	47.95	47.95	52.05	51.46	49.71	47.95	52.05	**54.97**
Carcinom	68.21	73.41	68.21	65.32	**74.57**	68.79	68.79	68.21	71.68	73.99
warpPIE10P	41.90	77.14	71.90	42.86	34.76	57.62	31.43	41.90	55.24	**81.43**
JAFFE	81.22	80.75	73.24	82.16	**84.51**	70.89	57.28	81.22	80.75	83.57
movement	46.94	50.28	45.28	45.00	50.83	50.00	49.17	48.06	50.00	**53.06**
NMI										
Digital	80.59	68.38	48.16	48.16	46.87	73.51	81.80	80.88	83.22	**83.96**
Flower17	50.19	45.14	45.35	44.12	50.73	52.13	53.89	50.42	57.11	**58.58**
ProteinFold	38.53	42.03	37.16	34.92	38.78	43.34	43.91	39.46	45.32	**46.93**
Flower102	46.32	48.99	42.30	43.24	48.17	54.18	53.36	49.73	57.27	**59.13**
Caltech102	59.90	59.07	59.64	55.17	55.86	58.20	58.40	59.90	60.38	**61.41**
warpAR10P	36.07	42.61	40.07	27.35	29.60	34.41	26.14	36.07	41.81	**50.55**
YALE	57.72	58.42	54.16	55.59	59.70	56.89	59.11	57.69	61.29	**62.27**
TOX171	27.15	26.57	26.93	27.15	31.23	28.51	**33.66**	27.15	27.15	28.91
Carcinom	68.32	72.79	71.48	68.32	69.76	67.39	67.90	68.32	70.85	**74.45**
warpPIE10P	49.20	78.67	76.23	49.73	39.95	60.64	33.28	49.20	59.18	**82.01**
JAFFE	78.25	78.53	72.73	79.43	**82.63**	71.07	64.41	78.25	78.01	81.15
movement	61.17	62.67	59.73	58.23	61.98	58.52	61.32	61.17	61.79	**64.23**
purity										
Digital	88.75	76.10	49.70	49.70	44.20	77.75	82.90	88.90	90.40	**91.05**
Flower17	51.99	44.63	46.84	45.81	51.54	56.47	53.24	53.01	61.03	**61.69**
ProteinFold	36.17	41.21	33.86	32.71	36.60	40.78	42.36	36.46	42.65	**45.24**
Flower102	32.28	38.78	27.61	28.79	33.86	44.08	40.24	38.87	46.39	**47.64**
Caltech102	37.12	35.10	37.25	29.41	31.70	35.75	33.27	37.12	37.65	**39.08**
warpAR10P	40.00	43.08	43.85	29.23	31.54	33.08	30.77	40.00	40.77	**47.69**
YALE	53.94	57.58	52.73	54.55	59.39	56.36	56.97	57.58	60.21	**61.82**
TOX171	48.54	47.95	47.95	48.54	52.05	51.46	50.29	49.12	52.05	**54.97**
Carcinom	72.25	76.88	77.46	75.14	76.30	73.99	73.99	72.25	76.30	**79.19**
warpPIE10P	44.76	78.10	71.90	44.76	37.62	60.48	32.38	44.76	56.19	**81.43**
JAFFE	81.22	81.22	75.59	82.16	**84.51**	71.36	58.22	81.22	80.75	83.57
movement	48.89	52.78	49.44	45.83	53.06	50.56	50.00	49.72	50.56	**53.61**

Table 3: ACC, NMI and purity comparison of different clustering algorithms on four bioinformatics data sets.

Datasets	A-MKKM	MKKM (Huang et al. 2012)	MKKM-MR (Liu et al. 2016)	Proposed
ACC				
psortPos	57.12	60.81	59.70	**64.33**
psortNeg	41.00	51.18	51.11	**53.60**
plant	61.70	56.38	61.38	**64.57**
nonplant	49.38	54.32	56.59	**59.57**
NMI				
psortPos	28.86	35.37	34.13	**44.33**
psortNeg	17.63	30.99	29.58	**32.78**
plant	26.82	20.02	26.41	**30.94**
nonplant	16.55	15.83	23.43	**26.04**
purity				
psortPos	60.81	66.91	65.25	**69.69**
psortNeg	43.14	55.47	53.95	**57.48**
plant	61.70	56.38	61.81	**64.57**
nonplant	72.18	71.45	75.33	**78.34**

experiment for 50 times with random initialization to reduce the affect of randomness caused by k-means, and report the best result.

Experimental results

The experimental study aims to verify the advantages and effectiveness of optimal neighborhood kernel clustering with multiple kernels. This study includes the following two parts: i) demonstrating the advantages of optimal neighborhood kernel clustering over the widely used linear combination of base kernels; and ii) showing the effectiveness of the proposed algorithm by conducting comprehensive experiments to compare with state-of-the-art clustering algorithms in the literature.

For the first part, to demonstrate the superiority of optimal neighborhood kernel clustering, we compare the A-MKKM, MKKM (Huang, Chuang, and Chen 2012) and MKKM-MR (Liu et al. 2016) with the proposed algorithm on four bioinformatics data sets. Note that the first three algorithms take the common assumption that the optimal kernel is a linear combination of base kernels. Differently, our algorithm relaxes this assumption and allows the optimal kernel to be

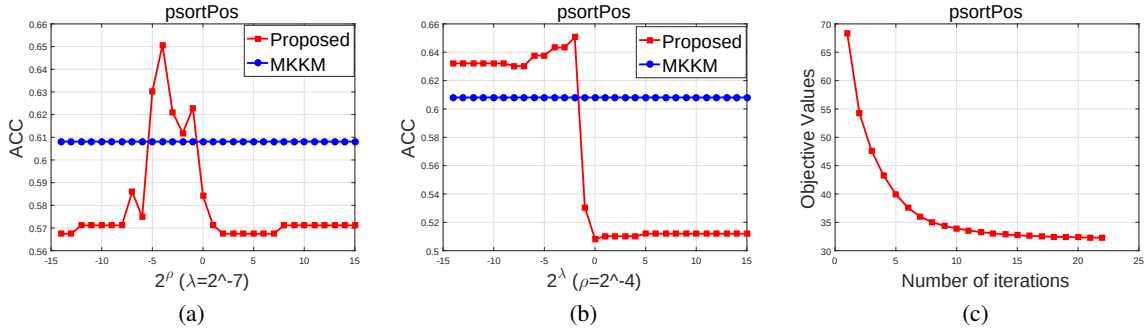

Figure 1: (a) The effect of ρ on clustering accuracy. (b) The effect of λ on clustering accuracy. (c) The objective value of our algorithm at each iteration.

resided in the neighborhood of a linear combination of base kernels.

The experimental results are reported in Table 3. As seen, on all datasets, our algorithm consistently and significantly outperforms MKKM-MR, which is considered as the state-of-the-art among MKKM based algorithms (Liu et al. 2016). In specific, our algorithm exceeds the second best one by 3.52%, 2.42%, 2.87% and 2.98% on psortPos, psortNeg, plant and nonpl in terms of ACC. Similar results can also be observed in terms of NMI and purity. This experiment clearly shows the advantages of optimal neighborhood kernel clustering. By assuming that the optimal kernel resides in the neighborhood of a linear combination of base kernels, our algorithm is able to search a more appropriate kernel in a large space for clustering. Meanwhile, the negotiation between the process of kernel learning and that of clustering makes the learned kernel better serve the clustering. Both factors contribute to the improvement of our algorithm.

For the second part, we conduct comprehensive experiments to compare our algorithm with several newly proposed ones on 12 benchmark data sets. The ACC, NMI and purity of the above-mentioned algorithms are reported in Table 2. From these results, we have the following observations:

- Our algorithm demonstrates the best clustering performance in terms of clustering accuracy, NMI and purity on most of the data sets. Taking the ACC as an example, it exceeds the second best one by 1.5%, 4.61%, 1.21%, 2.92%, 4.29% and 2.23% on Caltech102, warpAR10P, YALE, TOX171, warpPIE10P and movement, respectively.
- The proposed algorithm significantly outperforms exiting MKKM based algorithms, including MKKM, LMKKM and MKKM-MR, on all 12 data sets in terms of ACC, NMI and purity. This again validates the advantages and effectiveness of optimal neighborhood kernel clustering.
- As two strong baselines, A-MKKM and SB-KKM usually demonstrate comparable or even better performance than most of algorithms in comparison. However, our algorithm consistently and significantly outperforms these baselines on all data sets, which indicates its superiority in clustering performance.

From the above experiments, we conclude that the proposed algorithm: i) effectively enhances the representation capability of the learned kernel; and ii) is able to learning a better kernel for clustering by strengthening the negotiation between optimal kernel learning and clustering.

Parameter selection and convergence

The proposed algorithm introduces two regularization parameters λ and ρ to balance three terms in Eq.(7). We then experimentally show the effect of each parameter on the performance of our algorithm by fixing the other on psortPos.

Figure 1(a) plots the ACC of our algorithm by varying ρ in a large range $[2^{-15}, 2^{-13}, \cdots, 2^{15}]$ with $\lambda = 2^{-7}$. From this figure, we observe: i) with the increase of ρ, the ACC first increases to a maximum and then decreases, validating the effectiveness of optimal neighborhood kernel clustering; and ii) our algorithm shows stable performance across a wide range of ρ. Similarly, Figure 1(b) presents the ACC of our algorithm by varying λ from 2^{-15} to 2^{15} with $\rho = 2^{-4}$. Again, our algorithm demonstrates stable performance across a wide range of λ. These results indicate that the performance of our method is stable across a wide range of parameters.

An example of the objective value of our algorithm at each iteration is plotted in Figure 1(c). As observed from this example, the objective value is monotonically decreased and the algorithm quickly converges in less than thirty iterations.

Conclusions

This work proposes the optimal neighborhood kernel clustering with multiple kernels—a more flexible and effective algorithm which enhances the representability of the learned optimal kernel and strengthens the negotiation between the kernel learning and clustering. A three-step algorithm with proved convergence is designed to solve the resultant optimization problem. Comprehensive experimental results clearly demonstrates the superiority of our algorithm. In the future, we plan to extend our algorithm to a more general framework, and use it as a platform to revisit existing multiple kernel clustering algorithms and uncover their relationship.

References

Bezdek, J. C., and Hathaway, R. J. 2003. Convergence of alternating optimization. *Neural, Parallel Sci. Comput.* 11(4):351–368.

Cortes, C.; Mohri, M.; and Rostamizadeh, A. 2012. Algorithms for learning kernels based on centered alignment. *JMLR* 13:795–828.

Du, L.; Zhou, P.; Shi, L.; Wang, H.; Fan, M.; Wang, W.; and Shen, Y.-D. 2015. Robust multiple kernel k-means clustering using ℓ_{21}-norm. In *IJCAI*, 3476–3482.

Gönen, M., and Margolin, A. A. 2014. Localized Data Fusion for Kernel k-Means Clustering with Application to Cancer Biology. In *NIPS*, 1305–1313.

Huang, H.; Chuang, Y.; and Chen, C. 2012. Multiple kernel fuzzy clustering. *IEEE T. Fuzzy Systems* 20(1):120–134.

Kumar, A., and Daumé, H. 2011. A co-training approach for multi-view spectral clustering. In *ICML*, 393–400.

Li, M.; Liu, X.; Wang, L.; Dou, Y.; Yin, J.; and Zhu, E. 2016. Multiple kernel clustering with local kernel alignment maximization. In *IJCAI*, 1704–1710.

Liu, J.; Chen, J.; Chen, S.; and Ye, J. 2009. Learning the optimal neighborhood kernel for classification. In *IJCAI*, 1144–1149.

Liu, X.; Dou, Y.; Yin, J.; Wang, L.; and Zhu, E. 2016. Multiple kernel k-means clustering with matrix-induced regularization. In *AAAI*, 1888–1894.

Lu, Y.; Wang, L.; Lu, J.; Yang, J.; and Shen, C. 2014. Multiple kernel clustering based on centered kernel alignment. *Pattern Recognition* 47(11):3656 – 3664.

Xia, R.; Pan, Y.; Du, L.; and Yin, J. 2014. Robust multi-view spectral clustering via low-rank and sparse decomposition. In *AAAI*, 2149–2155.

Yu, S.; Tranchevent, L.-C.; Liu, X.; Glänzel, W.; Suykens, J. A. K.; Moor, B. D.; and Moreau, Y. 2012. Optimized data fusion for kernel k-means clustering. *IEEE TPAMI* 34(5):1031–1039.

Zhao, B.; Kwok, J. T.; and Zhang, C. 2009. Multiple kernel clustering. In *SDM*, 638–649.

Zhou, P.; Du, L.; Shi, L.; Wang, H.; and Shen, Y.-D. 2015. Recovery of corrupted multiple kernels for clustering. In *IJCAI*, 4105–4111.