Robust Multiple Kernel K-means using $\ell_{2,1}$-norm

Liang Du[1,2], **Peng Zhou**[1,3], **Lei Shi**[1,3], **Hanmo Wang**[1,3], **Mingyu Fan**[4], **Wenjian Wang**[2*], **Yi-Dong Shen**[1†]

[1]State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences
[2]School of Computer and Information Technology, Shanxi University
[3]University of Chinese Academy of Sciences
[4]Institute of Intelligent System and Decision, Wenzhou University
{duliang,zhoup,ydshen}@ios.ac.cn, fanmingyu@amss.ac.cn, wjwang@sxu.edu.cn

Abstract

The k-means algorithm is one of the most often used method for data clustering. However, the standard k-means can only be applied in the original feature space. The kernel k-means, which extends k-means into the kernel space, can be used to capture the non-linear structure and identify arbitrarily shaped clusters. Since both the standard k-means and kernel k-means apply the squared error to measure the distances between data points and cluster centers, a few outliers will cause large errors and dominate the objection function. Besides, the performance of kernel method is largely determined by the choice of kernel. Unfortunately, the most suitable kernel for a particular task is often unknown in advance. In this paper, we first present a robust k-means using $\ell_{2,1}$-norm in the feature space and then extend it to the kernel space. To recap the powerfulness of kernel methods, we further propose a novel robust multiple kernel k-means (RMKKM) algorithm that simultaneously finds the best clustering label, the cluster membership and the optimal combination of multiple kernels. An alternating iterative schema is developed to find the optimal value. Extensive experiments well demonstrate the effectiveness of the proposed algorithms.

1 Introduction

Clustering is one of the most important research topics in both machine learning and data mining communities. It aims at partitioning the data points into groups with similar objects. An enormous number and variety of methods have been proposed over the past decades. Due to the simplicity and the effectiveness, the k-means algorithm [Wu *et al.*, 2008] becomes one of the most commonly-used methods for data clustering. However, the standard k-means algorithm is limited to the squared Euclidean distance and cannot identify arbitrarily shaped clusters.

Kernel clustering algorithms have the ability to capture the non-linear structure inherent in many real world data sets and thereby usually achieve better clustering performance. The typical kernel-based clustering methods include: kernel k-means [Schölkopf *et al.*, 1998] and spectral clustering [Ng *et al.*, 2002]. Efforts have been made to further improve the result of kernel k-means. [Dhillon *et al.*, 2007] shows that the spectral clustering can be equivalently reformulated as a weighted variant of kernel k-means. [Yu *et al.*, 2012] and [Huang *et al.*, 2012b] propose to integrate multiple information for clustering. [Chitta *et al.*, 2011], [Chitta *et al.*, 2012] and [Elgohary *et al.*, 2014] develop efficient optimization schema to improve the scalability of kernel k-means to hanle large scale data.

Although kernel k-means based methods have attracted increasing attention in recent years, they still suffer from the following problems in real applications. First, these methods adopt the squared loss, which is well-known to be unstable with respect to outliers and noises, to measure the reconstruction error. As a result, a few noisy entries with large errors may easily dominate the result. Second, it is known that the performance of these single kernel methods are largely determined by the choice of kernel. Unfortunately, the most suitable kernel for a particular task is often unknown in advance. Moreover, exhaustive search on a user-defined pool of kernels will be quite time-consuming when the size of the pool becomes large [Zeng and Cheung, 2011]. In addition, real world data sets are often generated from different sources equipped with heterogeneous features. However, single kernel methods may fail to fully utilize such information. By leveraging multiple input kernels from different views or sources, multiple kernel methods are with great potential to integrate complementary information [Yu *et al.*, 2012]. To alleviate the effort for kernel designing and make full use of complementary information, it is imperative to learn an appropriate kernel efficiently to make the performance of the kernel k-means robust or even improved. However, unlike the multiple kernel learning in supervised setting [Gönen and Alpaydın, 2011], kernel learning in unsupervised scenario is more challenging due to the absence of class labels that could guide the search for relevant kernels.

To improve the robustness with respect to noises and outliers, we first propose a robust k-means method by using $\ell_{2,1}$-norm to measure the distances between data points and clus-

ter centers. To recap the powerfulness of kernel method, we also provide the robust k-means in kernel space. To alleviate the effort of kernel construction and make use of complementary information, we further propose a robust multiple kernel k-means (RMKKM) algorithm. In particular, RMKKM performs robust k-means with an appropriate consensus kernel learned from a linear combination of multiple input kernels. To solve the optimization problem where all the variables are encapsulated by the non-smooth $\ell_{2,1}$-norm, an alternating iterative algorithm is provided to find the optimal value. Experimental results on benchmark data sets have shown that the proposed approaches achieve better clustering results in both the single kernel and multiple kernel learning settings.

The rest of the paper is organized as follows. The robust multiple kernel k-means is introduced in Section 2. Section 3 provides the optimization schema. A variety of experimental results are presented in Section 4. Finally, we provide some concluding remarks in Section 5.

Notations. In this paper, matrices are written as uppercase letters and vectors are written as boldface lower-case letters. Given a matrix $A = \{a_{ij}\}$, we denote its i-th column as $\boldsymbol{a}_i$. The ℓ_2-norm of a vector $\boldsymbol{x}$ is defined as $||\boldsymbol{x}||_2 = \sqrt{\boldsymbol{x}^T \boldsymbol{x}}$. The $\ell_{2,1}$-norm of a matrix is defined as $||A||_{2,1} = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{m} a_{ij}^2} = \sum_{i=1}^{n} ||\boldsymbol{a}_i||_2$.

2 Robust Multiple Kernel K-means

In this section, we first present the robust k-means using the $\ell_{2,1}$-norm. Then we extend it to kernel space and get robust multiple kernel k-means. To improve the effectiveness of kernel method, we further propose a novel robust multiple kernel k-means (RMKKM) algorithm for data clustering.

2.1 Robust K-means

Suppose we are given a data set $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n] \in \mathcal{R}^{d \times n}$, and aim to partition these data points into c disjoint clusters $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_c\}$. The k-means algorithm finds optimal solutions with respect to the clustering error, defined as the sum of squared Euclidean distance between each data point $\boldsymbol{x}_i$ and the cluster center $\boldsymbol{u}_j$ where $\boldsymbol{x}_i$ belongs to. The objective function of k-means can be written as

$$\min_{Z,U} \quad ||X - UZ^T||_2^2 \tag{1}$$

$$\text{s.t.} \quad z_{ij} = \{0,1\}, \sum_{j=1}^{c} z_{ij} = 1, \forall i = 1, 2, \cdots, n,$$

where $U \in \mathcal{R}^{d \times c}$ is the cluster centroid matrix, and $Z \in \mathcal{R}^{n \times c}$ is the cluster indicator matrix.

Though the k-means algorithm has been widely used for data clustering, it is known that the squared loss used in Eq. (1) is very sensitive to data outliers, and they greatly affect the performance of clustering. In order to have a more stable result with respect to a fixed initialization, the robust k-means algorithm is desired. Inspired by the recent developed robust multi-view k-means [Cai *et al.*, 2013], we also use the $\ell_{2,1}$-norm to measure the reconstruction error, and get

the following problem

$$\min_{Z,U} \quad ||X^T - ZU^T||_{2,1} = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{c} z_{ij}||\boldsymbol{x}_i - \boldsymbol{u}_j||^2} \tag{2}$$

$$\text{s.t.} \quad z_{ij} = \{0,1\}, \sum_{j=1}^{c} z_{ij} = 1, \forall i = 1, 2, \cdots, n,$$

where the ℓ_1-norm is imposed among data points and the ℓ_2-norm is used for features. By introducing the sparsity induced norm, $\ell_{2,1}$-norm, the effects of data outliers can be reduced and a more robust result is expected.

2.2 Robust Kernel K-means

The effectiveness of k-means is largely limited to spherical clusters, that is, the clusters must be linearly separable [Dhillon *et al.*, 2004]. By applying kernel trick, the kernel k-means [Schölkopf *et al.*, 1998] algorithm attempts to address this problem by mapping data with non-linear transformation into appropriate feature space and applying k-means on the induced feature space. This can lead to linear separated clusters in the new feature space while these clusters are not linear separable in the original space. Thus, kernel k-means is much preferred for general clusteringn [Bishop and others, 2006].

Similar to the kernel k-means algorithm, the proposed robust k-means in Eq. (2) can also be easily extended to the kernel version by using a general kernelization framework [Zhang *et al.*, 2010]. Let $\phi : \mathcal{R}^d \rightarrow \mathcal{H}$ be a kernel mapping from the original space to the kernel space, where $\mathcal{H}$ is a Reproducing Kernel Hilbert Space (RKHS) induced by a kernel function $\mathcal{K}(\boldsymbol{x}, \boldsymbol{z}) = \phi(\boldsymbol{x})^T \phi(\boldsymbol{z})$. Then $\phi(X) = [\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2), \cdots, \phi(\boldsymbol{x}_n)]$, and the robust k-means in Eq. (2) in the kernel space becomes:

$$\min_{Z,V} \quad ||\phi(X)^T - ZV^T||_{2,1} \tag{3}$$

$$= \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{c} z_{ij}||\phi(\boldsymbol{x}_i) - \boldsymbol{v}_j||^2},$$

where V is the cluster center matrix in the implicit feature space. It can be shown later (see Eq. (11)), the j-th cluster center $\boldsymbol{v}_j$ can be determined by $\boldsymbol{v}_j = \sum_i a_{ij}\phi(\boldsymbol{x}_i)$, where a_{ij} is the membership of $\boldsymbol{x}_i$ belongs to j-th cluster. Hence, the above problem can be re-written as

$$\min_{Z,A} \quad \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{c} z_{ij}(k_{ii} - 2\boldsymbol{a}_j^T \boldsymbol{k}_i + \boldsymbol{a}_j^T K \boldsymbol{a}_j)}, \tag{4}$$

where the kernel matrix is denoted by $K \in \mathcal{R}^{n \times n}$ and the (i, j)-th entry of K is $k_{ij} = \phi(\boldsymbol{x}_i)^T \phi(\boldsymbol{x}_j)$.

2.3 Robust Multiple Kernel K-means

The proposed method above only works for single kernel data clustering. However, one of the central problems with kernel methods in general is that it is often unclear which kernel is the most suitable for a particular task. In this subsection, we

further extend the above method to automatically learn an appropriate kernel from the convex linear combination of several pre-computed kernel matrices within the multiple kernel learning framework [Gönen and Alpaydın, 2011].

Suppose there are altogether m different kernel functions $\{\mathcal{K}\}_{t=1}^m$ available for the clustering task in hand. Accordingly, there are m different associated feature spaces denoted as $\{\mathcal{H}\}_t^m$. To combine these kernels and also ensure that the resulted kernel still satisfies Mercer condition, we consider a nonnegative combination of these feature maps, ϕ', that is,

$$\phi'(\boldsymbol{x}) = \sum_{t=1}^m w_t \phi_t(\boldsymbol{x}) \quad \text{with } w_t \geq 0. \tag{5}$$

Unfortunately, as these implicit mappings do not necessarily have the same dimensionality, such a linear combination may be unrealistic. Hence, we construct an augmented Hilbert space $\tilde{\mathcal{H}} = \oplus_{t=1}^m \mathcal{H}^i$ by concatenating all feature spaces $\tilde{\phi}(\boldsymbol{x}) = [\sqrt{w_1}\phi_1(\boldsymbol{x}); \sqrt{w_2}\phi_2(\boldsymbol{x}); \dots; \sqrt{w_m}\phi_m(\boldsymbol{x})]^T$ with different weight $\sqrt{w_t}(w_t \geq 0)$, or equivalently the importance factor for kernel function $\mathcal{K}^t$. It can be verified that clustering in feature space $\tilde{\mathcal{H}}$ is equivalent to employing a combined kernel function [Zeng and Cheung, 2011]

$$\tilde{\mathcal{K}}(\boldsymbol{x}, \boldsymbol{z}) = \sum_{t=1}^m w_t \mathcal{K}^t(\boldsymbol{x}, \boldsymbol{z}). \tag{6}$$

It is known that the convex combination, with $\boldsymbol{w}(w_t \geq 0)$, of the positive semi-definite kernel matrices $\{K\}_{t=1}^m$ is still a positive semi-definite kernel matrix. By replacing the single kernel in Eq. (4) with the combined kernel, we propose a new Robust Multiple Kernel k-means method by solving:

$$\min_{Z,A,\boldsymbol{w}} \quad \sum_{i=1}^n \sqrt{\sum_{j=1}^c z_{ij}[\sum_{t=1}^m w_t(k_{ii}^t - 2\boldsymbol{a}_j^T \boldsymbol{k}_i^t + \boldsymbol{a}_j^T K^t \boldsymbol{a}_j)]}$$
$$= \sum_{i=1}^n \sqrt{\sum_{j=1}^c z_{ij}(\tilde{k}_{ii} - 2\boldsymbol{a}_j^T \tilde{\boldsymbol{k}}_i + \boldsymbol{a}_j^T \tilde{K} \boldsymbol{a}_j)} \tag{7}$$
$$\text{s.t.} \quad z_{ij} = \{0,1\}, \sum_{j=1}^c z_{ij} = 1, \sum_{t=1}^m w_t^\gamma = 1, w_t \geq 0,$$

where γ is the parameter to control the kernel weight distribution with $0 < \gamma < 1$. It should be noticed that RMKKM aims at improving the stability of kernel k-means without suffering the choice of appropriate kernel functions, where the weak kernels can be computed either on single view or multi-view data [Cai et al., 2013].

3 Optimization Algorithm and Analysis

The optimization problem in Eq. (7) is not convex in all variables together, but convex in them separately. The difficulty of solving the proposed objective mainly lies that all variables are encapsulated into a non-smooth $\ell_{2,1}$-norm. In the following, we introduce an iterative algorithm based on block coordinate descent to solve it. We separately update the value of A, Z, and $\boldsymbol{w}$ while holding the other variables as constant. Thus, a local minima can be expected by solving a sequence of convex optimization problems.

3.1 Optimizing w.r.t. cluster centroid when Z and $\boldsymbol{w}$ are fixed

Recall that, we have introduced cluster centroid matrix V in the implicit feature space in Eq. (3), conceptually we further denote $\tilde{\boldsymbol{v}}_j$ to be the concatenated implicit representations for j-th cluster center. The optimization problem in Eq. (7) with respect to the cluster centroid matrix can be re-written as

$$\mathcal{J}(\tilde{V}) = \sum_{i=1}^n \sqrt{\sum_{j=1}^c z_{ij}||\tilde{\phi}(\boldsymbol{x}_i) - \tilde{\boldsymbol{v}}_j||^2}. \tag{8}$$

It can be verified that the derivative of $\mathcal{J}$ is equivalent to the derivative of the following proxy function $\mathcal{L}(\tilde{V})$,

$$\mathcal{L}(\tilde{V}) = \sum_{i=1}^n D_{ii} \sum_{j=1}^c z_{ij}(\tilde{\phi}(\boldsymbol{x}_i) - \tilde{\boldsymbol{v}}_j)^2, \tag{9}$$

where $D_{ii} = \frac{1}{2\sqrt{\sum_{j=1}^c z_{ij}||\tilde{\phi}(\boldsymbol{x}_i) - \tilde{\boldsymbol{v}}_j||^2}}$, conceptually.

Actually, the above problem is the weighted version of kernel k-means [Dhillon et al., 2004]. By setting the derivative to be zero, we have

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{v}_j} = -2 \sum_{i=1}^n z_{ij}(\tilde{\phi}(\boldsymbol{x}_i) - \tilde{\boldsymbol{v}}_j) = 0. \tag{10}$$

$$\boldsymbol{v}_j = \frac{\sum_{i=1}^n z_{ij} D_{ii} \tilde{\phi}(\boldsymbol{x}_i)}{\sum_{i=1}^n z_{ij} D_{ii}}. \tag{11}$$

Thus, it is clear that the cluster centroid matrix for robust (multiple) kernel k-means can be determined by the weighted combination of the data in implicit feature space, where the membership matrix A can be computed by

$$a_{ij} = \frac{z_{ij} D_{ii}}{\sum_{i=1}^n z_{ij} D_{ii}}. \tag{12}$$

And the actual value of D_{ii} can be computed by

$$D_{ii} = \frac{1}{2\sqrt{\sum_{j=1}^c z_{ij} \sum_{t=1}^m w_t(k_{ii}^t - 2\boldsymbol{a}_j^T \boldsymbol{k}_i^t + \boldsymbol{a}_j^T K^t \boldsymbol{a}_j)}}. \tag{13}$$

3.2 Optimizing w.r.t. partition matrix Z when A and $\boldsymbol{w}$ are fixed

The minimization of the objective function in Eq. (7) with respect to Z can be decomposed into solving n independent sub-problems, that is

$$\min_{\boldsymbol{z}_i} \quad \sqrt{\sum_{j=1}^c z_{ij}(\tilde{k}_{ii} - 2\boldsymbol{a}_j^T \tilde{\boldsymbol{k}}_i + \boldsymbol{a}_j^T \tilde{K} \boldsymbol{a}_j)}, \tag{14}$$

where $z_{ij} = \{0,1\}, \sum_{j=1}^c z_{ij} = 1$. And, the optimal solution of the above problem is given by

$$z_{ij} = \begin{cases} 1 & j = \arg\min_{j'}(\sum_{t=1}^m w_t(\boldsymbol{a}_{j'}^T K^t \boldsymbol{a}_{j'} - 2\boldsymbol{a}_{j'}^T \boldsymbol{k}_i^t)) \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

3.3 Optimizing w.r.t. kernel weight w when A and Z are fixed

By defining $E \in \mathcal{R}^{n \times m}$ with

$$e_{it} = \sum_{j=1}^{c} z_{ij}(k_{ii}^t - 2\boldsymbol{a}_j^T \boldsymbol{k}_i^t + \boldsymbol{a}_j^T K^t \boldsymbol{a}_j), \qquad (16)$$

the optimization of Eq. (7) with respect to $\boldsymbol{w}$ can be simplified as solving the following problem

$$\min \quad \sum_{i=1}^{n} \sqrt{\sum_{t=1}^{m} \boldsymbol{w}_t e_{it}}, \quad \text{s.t.} \sum_{t=1}^{m} w_t^\gamma = 1, w_t \geq 0. \quad (17)$$

It is obvious that the derivative in Eq. (17) can also be regarded as the derivative of the following objective function

$$\mathcal{L}(\boldsymbol{w}) = \sum_{t=1}^{m} w_t h_t, \quad \text{s.t.} \quad \sum_{t=1}^{m} w_t^\gamma = 1, w_t \geq 0, \quad (18)$$

where h_t is denoted as

$$h_t = \sum_{i=1}^{n} \frac{e_{it}}{2\sqrt{\sum_{t'=1}^{m} w_{t'} e_{it'}}}. \qquad (19)$$

The Lagrange function of Eq. (18) is $\mathcal{J}(\boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{h} + \lambda(1 - \sum_t w_t^\gamma)$. By using the KKT condition $\frac{\partial \mathcal{J}(\boldsymbol{w})}{\partial w_t} w_t = 0$ and the constraint $\sum_{t=1}^{m} w_t^\gamma = 1$, the optimal solution of $\boldsymbol{w}$ can be obtained by

$$w_t = \left[\frac{1}{\sum_{t'=1}^{m} h_{t'}^{\frac{\gamma}{\gamma-1}}} \right]^{\frac{1}{\gamma}} h_t^{\frac{1}{\gamma-1}}. \qquad (20)$$

In summary, we present the alternating iterative algorithm for optimizing Eq. (7) in Algorithm 1.

Algorithm 1 The algorithm of RMKKM

Input: A set of kernels $\{K^t\}_{t=1}^{m}$, the desired number of cluster c and the parameter γ.
 Initialize the indicator matrix Z randomly, such that Z satisfies $z_{ij} = \{0, 1\}$ and $\sum_j z_{ij} = 1$;
 Initialize the kernel weight $w_t = 1/m$ for each kernel;
 Initialize the diagonal matrix $D = I_n$, where I_n is the identity matrix;
 repeat
  Update the membership matrix A by Eq. (12);
  Update the cluster indicator matrix Z by Eq. (15);
  Update the kernel weight $\boldsymbol{w}$ by Eq. (20);
  Update the diagonal matrix D by Eq. (13);
 until Converges
Output: Partition matrix Z, membership matrix A, and the kernel weights $\boldsymbol{w}$.

3.4 Convergence Analysis

The optimization problem of Eq. (7) is a convex problem with respect to one variable while holding the others. For each subproblem, our algorithm will guarantee that we can find the optimal solution. Therefore, by solving these subproblems alternatively, our algorithm will reduce the objective function monotonically. Meanwhile, the whole problem is lower bounded. Thus, the convergence of the proposed algorithm can be verified.

3.5 Complexity Analysis

In the following, we give the complexity analysis of the optimization algorithm. Initially, we need to compute the $n \times n \times m$ kernel matrices $\{K\}_{t=1}^{m}$, whose cost is generally $O(n^2dm)$. In each iteration, the cost of updating A by Eq. (12) is $O(nc)$, the cost of updating Z by Eq. (15) is $O((n^2 + n)cm)$, the cost of updating $\boldsymbol{w}$ by Eq. (20) is $O((n^2 + n)nm + nm + m)$, and the cost of updating D by Eq. (13) is $O((n^2 + n)nm)$. Since $c \ll n$, the total cost is $O(n^2dm + (n^3 + n^2 + n)ml)$, where l is the number of iterations. Compared with kernel k-means with fixed kernel weights and sample weights, the proposed algorithm is less efficient due to the involvement of updating kernel weight $\boldsymbol{w}$ and sample weight D.

3.6 Discussion of the parameter γ

The parameter γ is used to control the distribution of weights for different kernels. From Eq. (20), we can observe that when $\gamma \to 1$, we will assign 1 to the kernel whose h_t is the smallest and assign 0 to the weights of other kernels. And when $\gamma \to 0$, we will get equal weights. With this strategy, we can avoid the trivial solution of the kernel weights $\boldsymbol{w}$ and control the whole weights distribution flexibly.

4 Experimental Results

In this section, we evaluate the clustering performance of the proposed algorithm on a number of real world data sets.

4.1 Data Sets

We collect a variety of data sets, including 5 image data sets and 3 text corpora, most of which have been frequently used to evaluate the performance of different clustering algorithms. The statistics of these data sets are summarized in Table 1.

Table 1: Description of the data sets

	# instances	# features	# classes
YALE	165	1024	15
JAFFE	213	676	10
ORL	400	1024	40
AR	840	768	120
COIL20	1440	768	20
TR11	414	6429	9
TR41	878	7454	10
TR45	690	8261	10

4.2 Compared Algorithms

To demonstrate how the clustering performance can be improved by the proposed approaches, we compared the results of the following algorithms:

Table 2: Clustering results measured by Accuracy/NMI/Purity of the compared methods.

Data	Metrics	KKM-b	KKM-a	SC-b	SC-a	RKKM-b	RKKM-a	KKM-ew	SC-ew	RKKM-ew	MKKM	AASC	RMKKM*
YALE	Acc	0.4712	0.3897	**0.4942**	0.4052	0.4809	0.3971	0.4100	0.4973	0.4106	0.4570	0.4064	**0.5218**
	NMI	0.5134	0.4207	**0.5292**	0.4479	0.5229	0.4287	0.4571	0.5326	0.4601	0.5006	0.4683	**0.5558**
	Purity	0.4915	0.4112	**0.5161**	0.4306	0.4979	0.4174	0.4345	0.5148	0.4358	0.4752	0.4233	**0.5364**
JAFFE	Acc	0.7439	0.6709	0.7488	0.5403	**0.7561**	0.6798	0.6254	0.5376	0.6277	0.7455	0.3035	**0.8707**
	NMI	0.8013	0.7148	0.8208	0.5935	**0.8347**	0.7401	0.6962	0.5913	0.7017	0.7979	0.2722	**0.8937**
	Purity	0.7732	0.7013	0.7683	0.5656	**0.7958**	0.7182	0.6655	0.5643	0.6683	0.7683	0.3308	**0.8890**
ORL	Acc	0.5353	0.4593	**0.5796**	0.4665	0.5496	0.4688	0.4726	0.4810	0.4815	0.4751	0.2720	**0.5560**
	NMI	0.7343	0.6336	**0.7516**	0.6674	0.7423	0.6391	0.6757	0.6939	0.6845	0.6886	0.4377	**0.7483**
	Purity	0.5803	0.5042	**0.6145**	0.5120	0.5960	0.5146	0.5189	0.5233	0.5285	0.5140	0.3156	**0.6023**
AR	Acc	0.3302	0.3089	0.2883	0.2222	**0.3343**	0.3120	0.3185	0.2188	0.3184	0.2861	0.3323	**0.3437**
	NMI	0.6521	0.6064	0.5837	0.5605	**0.6544**	0.6081	0.6334	0.5805	0.6334	0.5917	0.6506	**0.6549**
	Purity	0.3552	0.3364	0.3324	0.2599	**0.3587**	0.3388	0.3464	0.2533	0.3464	0.3046	0.3498	**0.3678**
COIL20	Acc	0.5949	0.5074	**0.6770**	0.4365	0.6164	0.5189	0.5483	0.3694	0.5543	0.5482	0.3487	**0.6665**
	NMI	0.7405	0.6357	**0.8098**	0.5434	0.7463	0.6370	0.7072	0.4647	0.7098	0.7064	0.4187	**0.7734**
	Purity	0.6461	0.5530	**0.6992**	0.4683	0.6635	0.5634	0.5945	0.3980	0.6012	0.5895	0.3914	**0.6995**
TR11	Acc	0.5191	0.4465	0.5098	0.4332	**0.5303**	0.4504	0.4382	0.4651	0.4384	0.5013	0.4715	**0.5771**
	NMI	0.4888	0.3322	0.4311	0.3139	**0.4969**	0.3348	0.3504	0.3814	0.3506	0.4456	0.3939	**0.5608**
	Purity	0.6757	0.5632	0.5879	0.5023	**0.6793**	0.5640	0.5825	0.5464	0.5826	0.6548	0.5467	**0.7293**
TR41	Acc	0.5564	0.4634	**0.6352**	0.4480	0.5676	0.4680	0.4755	0.4723	0.4784	0.5610	0.4590	**0.6265**
	NMI	0.5988	0.4037	**0.6133**	0.3660	0.6077	0.4086	0.4245	0.4362	0.4292	0.5775	0.4305	**0.6347**
	Purity	0.7446	0.6000	0.7368	0.5645	**0.7499**	0.6021	0.6367	0.6269	0.6395	0.7283	0.6205	**0.7757**
TR45	Acc	**0.5879**	0.4558	0.5739	0.4596	0.5813	0.4569	0.4512	0.5329	0.4530	0.5846	0.5264	**0.6400**
	NMI	**0.5787**	0.3869	0.4803	0.3322	0.5786	0.3896	0.4022	0.4198	0.4057	0.5617	0.4194	**0.6273**
	Purity	**0.6849**	0.5364	0.6125	0.5002	0.6818	0.5375	0.5586	0.5697	0.5604	0.6914	0.5749	**0.7520**
Average	mAcc	0.5423	0.4627	**0.5634**	0.4264	0.5521	0.4690	0.4674	0.4468	0.4703	0.5198	0.3900	**0.6003**
	mNMI	0.6385	0.5168	0.6275	0.4781	**0.6480**	0.5233	0.5433	0.5126	0.5469	0.6088	0.4364	**0.6811**
	mPurity	0.6190	0.5257	0.6085	0.4754	**0.6279**	0.5320	0.5422	0.4996	0.5453	0.5908	0.4441	**0.6690**

* The parameter γ for RMKKM is set to 0.4 for all the data sets.

- Single kernel methods. Since we have multiple input kernels in hand, we run Kernel K-means (KKM), Spectral Clustering (SC) [Ng *et al.*, 2002], and Robust Kernel K-means (RKKM) in Eq. (4) on each kernel separately. And both the best and the average results over all these kernels are reported, which are referred to KKM-b, KKM-a, SC-b, SC-a, RKKM-b, RKKM-a, respectively. Due to space limitation, the worst results for single kernel are not reported. It should be pointed out that the worst results are often far below the average.

- Equal weighted methods. The multiple input kernels are combined into a single kernel with equal weights, and then report the results of KKM-ew, SC-ew, RKKM-ew. Note that such strategy often gets more stable results in multiple kernel learning [Gönen and Alpaydın, 2011].

- MKKM[1]. The MKKM proposed in [Huang *et al.*, 2012b] also extends k-means in multiple kernel setting. Both the loss function and the constraint on the kernel

weight distribution are different to our method.

- AASC[2]. The AASC algorithm is proposed in [Huang *et al.*, 2012a] for multiple affinities aggregation.

- RMKKM[3]. The proposed robust multiple kernel k-means method for data clustering.

4.3 Experiment Setup

Following the similar strategy of other multiple kernel learning approaches, we apply 12 different kernel functions as basis for multiple kernel clustering. These kernels include, seven RBF kernels $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-||\boldsymbol{x}_i - \boldsymbol{x}_j||^2/2\delta^2)$ with $\delta = t * D_0$, where D_0 is the maximum distance between samples and t varies in the range of $\{0.01, 0.05, 0.1, 1, 10, 50, 100\}$, four polynomial kernels $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = (a + \boldsymbol{x}_i^T \boldsymbol{x}_j)^b$ with $a = \{0, 1\}$ and $b = \{2, 4\}$ and a cosine kernel $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^T \boldsymbol{x}_j)/(||\boldsymbol{x}_i|| \cdot ||\boldsymbol{x}||)$. Final-

---

[1] http://imp.iis.sinica.edu.tw/IVCLab/research/Sean/mkfc/code.rar.

[2] http://imp.iis.sinica.edu.tw/IVCLab/research/Sean/aasc/code.rar.

[3] For the purpose of reproducibility, we provide the code at https://github.com/csliangdu/RMKKM.

ly, all the kernels have been normalized through $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)/\sqrt{\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_i)\mathcal{K}(\boldsymbol{x}_j, \boldsymbol{x}_j)}$ and then rescaled to $[0, 1]$.

The number of clusters is set to the true number of classes for all the data sets and clustering algorithms. For SC and AASC, the discrete clustering result is obtained by running k-means on spectral embedding. For the proposed method RMKKM, the parameter γ to control the kernel weight distribution is set to 0.4. In addition, the results of all these compared algorithms depend on the initialization. As suggested in [Yang *et al.*, 2010], we independently repeat the experiments for 20 times with random initializations and report the best results corresponding to the best objective values.

Three clustering evaluation metrics are adopted to measure the clustering performance, that is, Clustering Accuracy (Acc), Normalized Mutual Information(NMI) and Purity.

4.4 Experimental Results

Table 2 shows the clustering results in terms of accuracy, NMI and purity on all the data sets. Both the best results for single kernel methods and multiple kernel methods are highlighted in boldface. These experiments reveal a number of interesting points: 1)The performance of single kernel methods (the first 6 columns), i.e., KKM, SC, RKKM, is largely determined by the choice of kernel function. On the one hand, with proper kernel function, these methods usually present good results. On the other hand, their performances are significantly deteriorated on inappropriate kernels. Such observations also motivate the development of robust kernel k-means for multiple kernel learning. Besides, it can also be seen that the proposed RKKM give best results on 5 data sets with single input kernel. 2)Clustering on the equal weighted multiple kernel combination often gives more reasonable and reliable results. Thus, it could be expected that with much carefully designed kernel weighting schema, the performance of multiple kernel clustering can be further boosted. 3)With proper kernel weight learning schema, the multiple kernel clustering approaches usually improve the results over simple equally weighted combination. 4)The proposed RMKKM gives the best results on these data sets for multiple kernel clustering. The performance of RMKKM is usually better or close to the result on the best single kernel. Note that, RMKKM does not need perform exhaustive search on a predefined pool of kernels. Such results well demonstrate the superiority of our method.

4.5 Parameter Selection

Our proposed method RMKKM introduces the parameter γ to control the kernel weight distribution. The effect of the parameter γ has been discussed in previous section. Here, we evaluate it empirically. Figure 1 shows how the clustering result in terms of NMI varies with the parameter γ on two image data sets, JAFFE and ORL, and two document data sets, TR11 and TR41.

As we can see, the performance of RMKKM is very stable with respect to the parameter γ. Compared with the average performance of single kernel method, i.e., KKM-a, and the result of equally weighted kernel fusion, i.e., KKM-ew, the RMKKM achieves consistently good performance when γ varies from 0.3 to 0.9 on all four data sets.
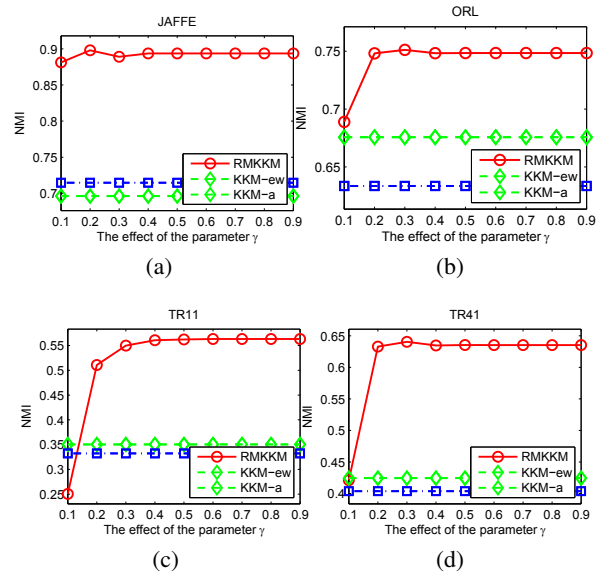

Figure 1: The performance of RMKKM is stable with respect to the parameter γ.

5 Conclusion

In this paper, we presented a robust k-means method which uses $\ell_{2,1}$-norm to quantify the distances between data points and cluster centers. Then we extended it to perform robust k-means in the kernel space. To fully recap the powerfulness of kernel methods, we further proposed a robust multiple kernel k-means algorithm for clustering, i.e. RMKKM. It automatically learns an appropriate kernel from a lot of input kernels, which significantly reduces the effort of kernel designing or selection. Experimental results well demonstrate the superiority of the proposed method on benchmark data sets with multiple input kernels.

It has been shown that, given the kernel weight, the remained problem can be finally reformulated as a adaptively weighted kernel k-means (see Eq. (9)). Thus, in the future, we plan to investigate the use of the Nyström method [Elgohary *et al.*, 2014] and the random fourier feature [Yang *et al.*, 2012] to make the proposed algorithms more efficient in terms of computational and memory complexity. Besides, we want to further investigate the effectiveness of the proposed method on multi-view and multi-source data clustering, where a bundle of input kernels is also available.

6 Acknowledgments

References

[Bishop and others, 2006] Christopher M Bishop et al. *Pattern recognition and machine learning*, volume 1. springer

New York, 2006.

[Cai *et al.*, 2013] Xiao Cai, Feiping Nie, and Heng Huang. Multi-view k-means clustering on big data. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2598–2604. AAAI Press, 2013.

[Chitta *et al.*, 2011] Radha Chitta, Rong Jin, Timothy C Havens, and Anil K Jain. Approximate kernel k-means: Solution to large scale kernel clustering. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 895–903. ACM, 2011.

[Chitta *et al.*, 2012] Radha Chitta, Rong Jin, and Anil K Jain. Efficient kernel clustering using random fourier features. In *ICDM*, pages 161–170, 2012.

[Dhillon *et al.*, 2004] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556. ACM, 2004.

[Dhillon *et al.*, 2007] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(11):1944–1957, 2007.

[Ding *et al.*, 2006] Chris Ding, Ding Zhou, Xiaofeng He, and Hongyuan Zha. R1-pca: rotational invariant l 1-norm principal component analysis for robust subspace factorization. In *Proceedings of the 23rd international conference on Machine learning*, pages 281–288. ACM, 2006.

[Elgohary *et al.*, 2014] Ahmed Elgohary, Ahmed K. Farahat, Mohamed S. Kamel, and Fakhri Karray. Embed and conquer: Scalable embeddings for kernel k-means on mapreduce. In *The SIAM International Conference on Data Mining (SDM)*, 2014.

[Gönen and Alpaydın, 2011] Mehmet Gönen and Ethem Alpaydın. Multiple kernel learning algorithms. *The Journal of Machine Learning Research*, 12:2211–2268, 2011.

[Gu and Zhou, 2009] Quanquan Gu and Jie Zhou. Co-clustering on manifolds. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 359–368. ACM, 2009.

[Huang *et al.*, 2012a] Hsin-Chien Huang, Yung-Yu Chuang, and Chu-Song Chen. Affinity aggregation for spectral clustering. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 773–780, 2012.

[Huang *et al.*, 2012b] Hsin-Chien Huang, Yung-Yu Chuang, and Chu-Song Chen. Multiple kernel fuzzy clustering. *Fuzzy Systems, IEEE Transactions on*, 20(1):120–134, 2012.

[Lee and Seung, 1999] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[Ng *et al.*, 2002] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.

[Schölkopf *et al.*, 1998] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[Wu *et al.*, 2008] Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, S Yu Philip, et al. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.

[Yang *et al.*, 2010] Yi Yang, Dong Xu, Feiping Nie, Shuicheng Yan, and Yueting Zhuang. Image clustering using local discriminant models and global integration. *Image Processing, IEEE Transactions on*, 19(10):2761–2773, 2010.

[Yang *et al.*, 2012] Tianbao Yang, Yu-Feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou. Nyström method vs random fourier features: A theoretical and empirical comparison. In *Advances in neural information processing systems*, pages 476–484, 2012.

[Yu *et al.*, 2011] Shi Yu, Léon-Charles Tranchevent, Bart De Moor, and Yves Moreau. *Kernel-based Data Fusion for Machine Learning - Methods and Applications in Bioinformatics and Text Mining*, volume 345 of *Studies in Computational Intelligence*. Springer, 2011.

[Yu *et al.*, 2012] Shi Yu, L-C Tranchevent, Xinhai Liu, Wolfgang Glanzel, Johan AK Suykens, Bart De Moor, and Yves Moreau. Optimized data fusion for kernel k-means clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(5):1031–1039, 2012.

[Zeng and Cheung, 2011] Hong Zeng and Yiu-ming Cheung. Feature selection and kernel learning for local learning-based clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1532–1547, 2011.

[Zhang *et al.*, 2010] Changshui Zhang, Feiping Nie, and Shiming Xiang. A general kernelization framework for learning algorithms based on kernel pca. *Neurocomputing*, 73(4):959–967, 2010.