



**HAL**  
open science

# A Novel Locally Multiple Kernel k-means Based on Similarity

Shuyan Fan, Shifei Ding, Mingjing Du, Xiao Xu

► **To cite this version:**

Shuyan Fan, Shifei Ding, Mingjing Du, Xiao Xu. A Novel Locally Multiple Kernel k-means Based on Similarity. 9th International Conference on Intelligent Information Processing (IIP), Nov 2016, Melbourne, VIC, Australia. pp.22-30, 10.1007/978-3-319-48390-0\_3. hal-01614989

**HAL Id: hal-01614989**

**<https://inria.hal.science/hal-01614989>**

Submitted on 11 Oct 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# A novel locally multiple kernel k-means based on similarity

Shuyan Fan<sup>1,2</sup>, Shifei Ding<sup>1,2</sup>, Mingjing Du<sup>1,2</sup>, Xiao Xu<sup>1,2</sup>

1 School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, 221116, China

2 Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China  
dingsf@cumt.edu.cn

**Abstract.** Most of multiple kernel clustering algorithms aim to find the optimal kernel combination and have to calculate kernel weights iteratively. For the kernel methods, the scale parameter of Gaussian kernel is usually searched in a number of candidate values of the parameter and the best is selected. In this paper, a novel multiple kernel k-means algorithm is proposed based on similarity measure. Our similarity measure meets the requirements of the clustering hypothesis, which can describe the relations between data points more reasonably by taking local and global structures into consideration. We assign to each data point a local scale parameter and combine the parameter with density factor to construct kernel matrix. According to the local distribution, the local scale parameter of Gaussian kernel is generated adaptively. The density factor is inspired by density-based algorithm. However, different from density-based algorithm, we first find neighbor data points using k nearest neighbor method and then find density-connected sets by union-find set method. Experiments show that the proposed algorithm can effectively deal with the clustering problem of datasets with complex structure or multiple scales.

**Keywords:** multiple kernel clustering, kernel k-means, similarity measure, clustering analysis

## 1 Introduction

Unsupervised data analysis using clustering algorithms provides a useful tool. The aim of clustering analysis is to discover the hidden data structure of a dataset according to a certain similarity criterion such that all the data points are assigned into a number of distinctive clusters where points in the same cluster are similar to each other, while points from different clusters are dissimilar [1]. Clustering has been applied in a variety of scientific fields such as web search, social network analysis, image retrieval, medical imaging, gene expression analysis, recommendation systems and market analysis and so on.

Kernel clustering method can handle data sets that are not linearly separable in input space [2], thus, usually perform better than the Euclidean distance based clustering algorithms [3]. Due to simplicity and efficiency, kernel k-means has become a hot research topic. The kernel function is used to map the input data into a high-dimensional feature space, which makes clusters that are not linearly separable in input space become separable. A single kernel is sometimes insufficient to represent the data.

Recently, multiple kernel clustering has gained increasing attention in machine learning. Huang et.al propose a multiple kernel fuzzy c-means [4]. By incorporating multiple kernels and automatically adjusting the kernel weights, ineffective kernels and irrelevant features are not crucial for kernel clustering. Zhou et al. use the maximum entropy method to regularize the kernel weights and decide the important kernels [5]. Gao applies multiple kernel fuzzy c-means to optimize clustering and presented mononuclear kernel function which is a set of Gaussian kernel function combination assigned different weights resolution [6]. Lu et al. applies multiple kernel k-means clustering algorithm into SAR image change detection [7]. They fuse various features through a weighted summation kernel by automatically and optimally computing the kernel weights, which leads to computational burden. Zhang et al. propose a locally multiple kernel clustering which assigns to each cluster a weight vector for feature selection and combines it with a Gaussian kernel to form a unique kernel for the corresponding cluster [8]. They search the scale parameter of Gaussian kernel by running their clustering algorithm repeatedly for a number of values of the parameter and selecting the best one. Tzortzis et al. overcome the kernel selection problem of maximum margin clustering by employing multiple kernel learning to jointly learn the kernel and a partitioning of the instances [9]. Yu et al. propose an optimized kernel k-means clustering which optimizes the cluster membership and kernel coefficients based on the same Rayleigh quotient objective [10]. Lu et al. improve kernel evaluation measure based on centered kernel alignment and their algorithm needs to be given the initial kernel fusion coefficients [11]. Although the above methods extend from different clustering algorithms, they all employ the alternating optimization technique to solve their extended problems. Specifically, cluster labels and kernel combination coefficients are alternatively optimized until convergence.

Our algorithm is proposed from perspective of similarity measure by calculating a local scale parameter for each data point, which can reflect local distribution of datasets. In addition, another parameter named density factor is introduced in Gaussian kernel function which can describe global structure of data set and avoid kernel k-means running into local optimum. Based on improved similarity measure, our algorithm has several advantages. First, as a kernel method, it has unusual ability in dealing with datasets with multiple scales. Second, it fuses automatically and optimally local and global structures of datasets. Furthermore, our algorithm does not need a good deal of iterations and calculate kernel weights until convergence.

The remainder of this paper is organized as follows: in Section 2 we introduce the related works. In Section 3 we give a detailed description of our algorithm. Section 4 presents the experimental results and evaluation of our algorithm. Finally, we conclude the paper in Section 5.

## 2 Related work

### 2.1 Kernel k-means

Girolami first proposed the kernel k-means clustering method. It first maps the data points from the input space to higher dimensional feature space through a nonlinear transformation  $\phi(\cdot)$  and then minimizes the clustering error in that feature space [12].

Let  $D = \{x_1, x_2, \dots, x_n\}$  be the data set of size  $n$ ,  $k$  be the number of clusters required. The final partition of the entire data set is  $\Pi_D = \{C_1, C_2, \dots, C_k\}$ . The objective function is to minimize the criterion function:

$$J = \sum_{j=1}^k \sum_{x_i \in C_j} \|\phi(x_i) - m_j\|^2 \quad (1)$$

Where  $m_j$  is the mean of cluster  $C_j$ . That is

$$m_j = \sum_{x_i \in C_j} \frac{\phi(x_i)}{|C_j|} \quad (2)$$

in the induced space.

$$\begin{aligned} \|\phi(x_i) - m_j\|^2 &= \left\| \phi(x_i) - \sum_{x_l \in C_j} \frac{\phi(x_l)}{|C_j|} \right\|^2 \\ &= \phi(x_i) \cdot \phi(x_i) + \frac{2}{|C_j|} \sum_{x_l \in C_j} \phi(x_l) \cdot \phi(x_i) + \frac{1}{|C_j|^2} \sum_{x_l \in C_j} \sum_{x_s \in C_j} \phi(x_l) \cdot \phi(x_s) \\ &= \kappa(x_i, x_i) + \frac{2}{|C_j|} \sum_{x_l \in C_j} \kappa(x_i, x_l) + \frac{1}{|C_j|^2} \sum_{x_l \in C_j} \sum_{x_s \in C_j} \kappa(x_l, x_s) \end{aligned} \quad (3)$$

Further,  $\|\phi(x_i) - m_j\|^2$  can be calculated without knowing the transformation  $\phi(\cdot)$  explicitly as formula (3).

Thus, only inner products are used in the computation of the Euclidean distance between a point and a centroid. If given a kernel matrix  $\kappa$ , where  $\kappa_{ij} = \phi(x_i) \cdot \phi(x_j)$ , A kernel function is commonly used to map the original points to inner products. Given a data set, the kernel k-means clustering has the following steps:

---

**Algorithm1.** Kernel k-means clustering algorithm.

---

**Input:**

Data set:  $D = \{x_1, x_2, \dots, x_n\}$

The number of cluster:  $k$

**Output:**

The final Partition:  $\Pi_D = \{C_1, C_2, \dots, C_k\}$ .

**Method:**

Step 1: Initialize the  $k$  clusters  $\{C_1^0, C_2^0, \dots, C_k^0\}$

Step 2: For each point  $x_i$  and each cluster  $C_i$ , compute  $\|\phi(x_i) - m_j\|^2$  using formula (3) and assign  $x_i$  to its nearest center.

Step 3: Update  $m_j$ , for  $j=1$  to  $k$  using formula (2).

Step 4: Repeat step 2 and 3 till convergence

---

## 2.2 Multiple Kernel k-means

Weighted summation kernel is a common tool for multiple kernel learning. Huang et al. propose multiple kernel k-means algorithm by incorporating weighted summation kernel into the kernel k-means, which results in the multiple kernel k-means algorithm [4]. The MKKM algorithm is solved by updating iteratively the kernel weights. Its objective function is to minimize

$$\begin{aligned} J_M &= \sum_{j=1}^k \sum_{x_i \in C_j} \sum_{m=1}^M w_m^2 \|\phi_k(x_i) - v_c\|^2 \\ w_m &= \frac{\frac{1}{\beta_m}}{\left(\frac{1}{\beta_1} + \frac{1}{\beta_2} + \dots + \frac{1}{\beta_M}\right)}, \beta_m = \sum_{j=1}^k \sum_{x_i \in C_j} \|\phi(x_i) - m_j\|^2 \end{aligned} \quad (4)$$

Where  $\{\phi_k\}_{m=1}^M$  are the mapping functions corresponding to multiple kernel functions.  $w_m (m = 1, 2, \dots, M)$  are kernel weights.

### 3 Locally multiple kernel k-means

#### 3.1 Similarity Measure

Selecting a suitable method of similarity measure in cluster analysis is crucial, and it is used as the basis for division [13]. To handle the dataset with multiple scales, we calculate a local scaling parameter  $\sigma_i$  for each data point  $s_i$ . The selection of the local scale  $\sigma_i$  can be done by studying the local statistics of the neighborhood of point  $s_i$ .  $s_K$  is the  $K$ 'th neighbor of point  $s_i$ .

$$\sigma_i = d(s_i, s_K)$$

According to the conception of clustering hypothesis, the data point of intra-class should locate in high-density region, and the data point of inter-class should be separated by low-density region [14]. In order to better describe global structure of data set and avoid kernel k-means running into local optimum, density factor  $\rho$  is introduced to discover clusters of arbitrary shape. Combined  $\rho$  with formula (6), we propose a new similarity measure as follows:

$$S_{ij} = \exp\left(\frac{-d^2(s_i, s_j)}{\sigma_i \sigma_j \rho_{ij}}\right) \quad (5)$$

Density factor is obtained by a simple and powerful way. First, find  $k$  neighbor points for each point by  $k$  nearest neighbor algorithm and then use  $k$  nearest neighbor graph to depict the local neighborhood relation between data points. The neighborhood of a point  $p$  is denoted by  $N(p)$ . For a sample point  $q$ , if  $q \in N(p)$ , we think  $q$  is directly density-reachable from point  $p$ . Given a sample set  $D = \{p_1, p_2, \dots, p_n\}$ , supposed that  $p_i$  is directly density-reachable from point  $p_{i+1}$ ,  $p_1$  is density-reachable from  $p_n$ . If there is a point  $o$  such that both,  $p$  and  $q$  are density-reachable from  $o$ , we consider that the point  $p$  is density-connected to a point  $q$ . Finally, according to all directly density-reachable data points, find density-connected sets by union-find set method which is a very sophisticated and practical data structures and mainly used for processing the merger of the problem of some disjoint sets [15]. Let  $\rho_{ij}$  denote density factor between the point  $s_i$  and  $s_j$ , as follows:

$$\rho_{ij} = \begin{cases} 1, & \text{if } s_i, s_j \text{ are in the same density-connected set} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

#### 3.2 Algorithm

From the perspective of similarity measure, we propose a novel locally multiple kernel k-means algorithm (LMKMM). Its basic idea is: firstly, calculate the local scale parameter  $\sigma$  and density factor  $\rho$ ; subsequently, construct kernel matrix based on our proposed similarity measure; finally, according to the kernel matrix, cluster dataset by kernel k-means. The detail steps of our algorithm are as follows.

---

**Algorithm2.** Locally multiple kernel k-means

---

**Input:**

Data set:  $D = \{x_1, x_2, \dots, x_n\}$

---

---

The number of cluster:  $k$

**Output:**

The final Partition:  $\Pi_D = \{C_1, C_2, \dots, C_k\}$ .

**Method:**

Step 1: Calculate distance matrix  $D$  using Euclidean distance.

Step 2: Calculate the  $k$  nearest neighborhoods and get  $k$  nearest neighbor graph.

Step 3: Find density-connected sets by union-find set method.

Step 4: According to density-connected sets, density factor is obtained by formula (6).

Step 5: Calculate similarity matrix according to formula (5).

Step 6: Initialize the  $k$  clusters  $\{C_1^0, C_2^0, \dots, C_k^0\}$ .

Step 7: For each point  $x_i$  and each cluster  $C_i$ , compute

$$d(x_i, m_j) = \kappa(x_i, x_i) + \frac{2}{|C_j|} \sum_{x_l \in C_j} \kappa(x_i, x_l) + \frac{1}{|C_j|^2} \sum_{x_l \in C_j} \sum_{x_s \in C_j} \kappa(x_l, x_s)$$

and assign  $x_i$  to its nearest center.

Step 8: Update  $m_j$ , for  $j=1$  to  $k$  using formula (2).

Step 9: Repeat step 7 and 8 till convergence.

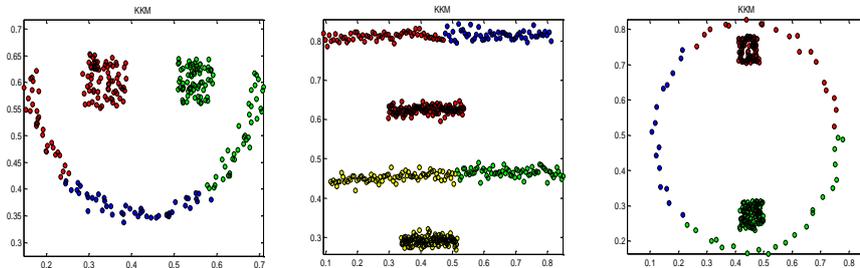
---

Suppose  $n$  is the total number of points on the data set. Our local multiple kernel k-means contains three main components: calculating parameters, constructing similarity matrix, and clustering. In the phase of calculating parameters, the complexity of  $k$  nearest neighbor algorithm and union-find set method both are  $O(n)$ . The complexity in calculating the similarity matrix is  $O(n^2)$ . At the last clustering phase, the complexity of  $k$ -means is  $O(n)$ . Our algorithm does not increase the complexity of kernel  $k$ -means, but it improves performance.

## 4 Experiments

### 4.1 Artificial data clustering

In order to verify the effectiveness of our improved algorithm, we choose three artificial data sets, “smile face”, “four lines” and “blobs and circle” to perform an experiment and compare with kernel  $k$ -means (KKM) algorithm.



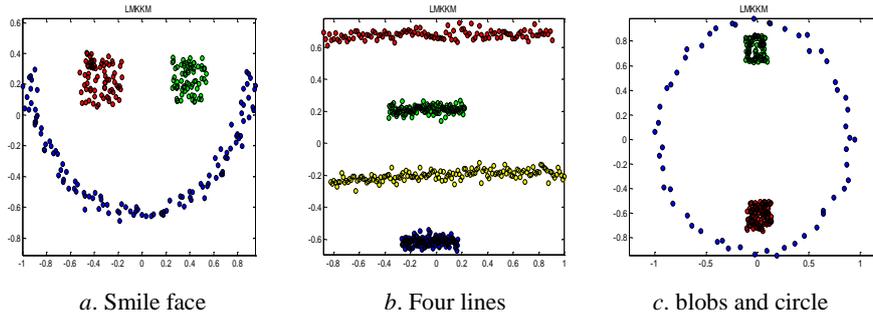
a. Smile face

b. Four lines

c. blobs and circle

**Figure 1.** KKM algorithm's clustering results on artificial data sets

Figure 1 shows KKM algorithm’s clustering results on artificial data sets. The scale parameter of Gaussian kernel function is set to be 1 experientially. KKM algorithm measures the similarity between points based on Euclidean distance which can not reflect the intrinsic structure of dataset. Thus, KKM can only gather the similar points in local region into a cluster, but does not satisfy the global coherence hypothesis of the clustering or recognize complex manifold structure of dataset.



**Figure 2.** LMKKM algorithm’s clustering results on artificial datasets

Figure 2 shows LMKKM algorithm’s clustering results on artificial datasets. It calculates the kernel matrix by formula (6). After the similarity measure involves density factor, it meets both the local coherence hypothesis and global coherence hypothesis of the clustering. Through the approach, the intra-class data points become more compact and the inner-class data points are more discrete.

#### 4.2 Clustering results

In this subsection, our method is compared with three baseline methods including kernel k-means (KKM), Self-Tuning Spectral Clustering (SSC) [16], locally adaptive multiple kernel clustering (LAMKC) [8]. SSC is a locally adaptive spectral clustering algorithm. LAMKC is a newly proposed multiple kernel clustering algorithm extending from kernel k-means. We carry out experiments on seven UCI datasets. These datasets are often used to test performance of machine learning algorithm. The characteristics of these data sets are shown in Table 1.

Table 1 Data characteristics of real data sets

Data set	Instance number	Attribute number	Cluster number
Iris	150	4	3
Sonar	208	60	2
WDBC	569	30	2
Ionosphere	351	34	2
Yeast	1484	8	10
Dermatology	366	34	6
Zoo	101	16	7

We use the accuracy ( $ACC$ ) to evaluate the clustering performance. Considering the random initialization of clustering centers of kernel k-means, clustering results will be fluctuated. Therefore the clustering experiments are repeated 20 times. Results in first row are the means of the 20 trials, and results in second row (in parentheses) are corresponding standard deviations. The neighborhood size of  $K$  is set to 7. For all experiments, the cluster number is set to be the cluster number of each dataset. For LAMKC, the stop condition of the gradient descent method is set to be 0.0001[8]. In

order to compare experimental results fairly, we do not use the Kaufman Approach to select a set of initial centroids in LAMKC. All experiments are conducted on Intel Pentium G2030 CPU with 3.00 GHz processor and 4 G RAM running 64bit-Win7. Clustering results are shown in Table 2.

**Table 2.** Clustering results on real-world data sets

Data set	KKM	SSC	LAMKC	LMKKM
Iris	0.8980 (0.1204)	0.9067 (0.0004)	0.9127 (0.0407)	<b>0.9600</b> (0.0312)
Sonar	0.5505 (0.0054)	0.5192 (0.0048)	0.6319 (0.0038)	<b>0.7337</b> (0.0325)
WDBC	0.5211 (0.1012)	0.8682 (0.0001)	<b>0.8832</b> (0.1426)	0.8714 (0.2308)
Ionosphere	0.5556 (0.0043)	0.5256 (0.0029)	0.7073 (0.0105)	<b>0.7620</b> (0.1704)
Yeast	0.3677 (0.0101)	0.3585 (0.2037)	0.4231 (0.0243)	<b>0.4502</b> (0.0041)
Dermatology	0.2268 (0.0164)	0.4426 (0.0004)	<b>0.6436</b> (0.1202)	0.5887 (0.0520)
Zoo	0.4109 (0.0589)	0.3953 (0.0184)	0.5208 (0.1331)	<b>0.6038</b> (0.0145)

The results on the seven UCI datasets are shown in Table 2. We use the boldface to mark the best result for each dataset. For the clustering performance measured by *ACC*, the experimental results are encouraging and our algorithm obtained five best results for the seven datasets. Comparable to the kernel k-means and self-tuning spectral clustering, performance of our algorithm is significantly better than that of them on seven datasets. For datasets WDBC and Dermatology, our algorithm is roughly comparable to LAMKC. In most cases, our algorithm can capture structures of datasets and calculate appropriate parameters adaptively while LAMKC searched the parameter of Gaussian kernel in a range. These indicate that our improved similarity measure has the capability to capture local and global structures of datasets with complexity so as that our algorithm can complete the tasks of clustering efficiently.

## 5 Conclusions

Conventional multiple kernel clustering algorithms aim to construct a global combination of multiple kernels in input space and have to kernel combination coefficients iteratively. In this paper, we proposed a local multiple kernel clustering method based on similarity measure. Our method is dedicated to the datasets with varying local distributions. Instead of using a uniform combination of multiple kernels over the whole input space, our method associates to each data point a localized kernel and combined with density factor simultaneously. Taking local and global structures

into consideration, our similarity measure can depict distributed situation of dataset. Results of clustering experiments on artificial datasets and UCI datasets demonstrate that our locally multiple kernel clustering method can deal with datasets with multiple scales and not fall into local optimal.

There are three points remaining for further research. First, the time complexity of our algorithm is the same as kernel k-means's, so it will spend much time when processing big data set. Further study is necessary on how to reduce the time complexity of the algorithm and improve the efficiency of clustering. Second, kernel k-means is sensitive to the initial cluster centers. We can improve kernel k-means from this perspective. Third, following the idea of this paper, we can construct better multiple kernel k-means methods based on the other kernel evaluation measures.

### **Acknowledgements**

This work is supported by the National Natural Science Foundation of China (No. 61379101, 61672522), and the National Key Basic Research Program of China (No. 2013CB329502).

### **References**

1. Ding, S., Zhang, J., Jia, H., et al.: An Adaptive Density Data Stream Clustering Algorithm. *Cognitive Computation*, 8(1): 30-38 (2016)
2. Chitta, R.: Kernel-based clustering of big data. *Dissertations & Theses – Gradworks* (2015)
3. Chitta, R., Jin, R., Havens, T.C., Jain, A.K.: Scalable Kernel Clustering: Approximate Kernel k-means. *Eprint Arxiv* (2014)
4. Huang, H.C., Chuang, Y.Y., Chen, C.S.: Multiple Kernel Fuzzy Clustering. *IEEE Transactions on Fuzzy Systems*, 20(1): 120-134 (2012)
5. Zhou, J., Chen, C.L., Chen, L., Maximum-entropy-based multiple kernel fuzzy c-means clustering algorithm. *IEEE International Conference on Systems, Man and Cybernetics IEEE* (2014)
6. Gao, S.: The Application of Clustering Optimization in Data Mining Based on Multiple Kernel Function FCM. *Journal of Computational Information Systems*, 11(11): 3977-3986 (2015)
7. Lu, J., Ming, L., Zhang, P., et al.: SAR Image Change Detection Based on Multiple Kernel K-Means Clustering With Local-Neighborhood Information. *IEEE Geoscience and Remote Sensing Letters*, pp.1-5 (2016)
8. Zhang, L., Hu, X.: Locally adaptive multiple kernel clustering. *Neurocomputing*, 137(11): 192-197 (2014)
9. Tzortzis, G., Likas, A.: Ratio-Based Multiple Kernel Clustering. *Machine Learning and Knowledge Discovery in Databases*, pp.241-257 (2014)
10. Yu, S., Tranchevent, L., Moor, B.D., et al.: Optimized Data Fusion for Kernel k-means Clustering. *Pattern Analysis & Machine Intelligence IEEE Transactions on*, 35(5): 1031-1039 (2011)
11. Lu, Y., Wang, L., Lu, J., et al.: Multiple kernel clustering based on centered kernel alignment, 47(11): 3656-3664 (2014)
12. Girolami, M.: Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, 13(3): 780–784 (2002)
13. Yan, J., Cheng, D., Zong, M., et al.: Improved Spectral Clustering Algorithm Based on Similarity Measure. *Advanced Data Mining and Applications*, Springer

International Publishing, pp.641-654 (2014)

14. Jia, H., Ding, S., Meng, L., et al.: A density-adaptive affinity propagation clustering algorithm based on spectral dimension reduction. *Neural Computing & Applications*, 25(7-8): 1557-1567 (2014)
15. Kaplan, H., Shafir, N., Tarjan, R.E.: Union-Find With Deletions. *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pp.19-28 (2002)
16. Manor, M.L.: Self-Tuning Spectral Clustering. *Advances in Neural Information Processing Systems*, 17: 1601—1608 (2004)