

# Performance Assessment of Kernel-Based Clustering

Meena Tushir and Smriti Srivastava

**Abstract** Kernel methods are ones that, by replacing the inner product with positive definite function, implicitly perform a nonlinear mapping of input data into a high-dimensional feature space. Various types of kernel-based clustering methods have been studied so far by many researchers, where Gaussian kernel, in particular, has been found to be useful. In this paper, we have investigated the role of kernel function in clustering and incorporated different kernel functions. We discussed numerical results in which different kernel functions are applied to kernel-based hybrid c-means clustering. Various synthetic data sets and real-life data set are used for analysis. Experiments results show that there exist other robust kernel functions which hold like Gaussian kernel.

**Keywords** Clustering · Kernel function · Gaussian kernel · Hyper-tangent kernel · Log kernel

## 1 Introduction

Fuzzy clustering has emerged as an important tool for discovering the structure of data. Kernel methods have been applied to fuzzy clustering, and the kernelized version is referred to as kernel-based fuzzy clustering. The kernel-based classification in the feature space not only preserves the inherent structure of groups in the input space, but also simplifies the associated structure of the data [1]. Since

---

M. Tushir (✉)

Maharaja Surajmal Institute of Technology, New Delhi, India  
e-mail: meenatushir@yahoo.com

S. Srivastava

Netaji Subash Institute of Technology, New Delhi, India  
e-mail: ssmriti@yahoo.com

Girolami first developed the kernel k-means clustering algorithm for unsupervised classification [2], several studies have demonstrated the superiority of kernel clustering algorithms over other approaches to clustering [3–5]. The point raised regarding the kernel-based clustering method of data partitioning is the choice of the type of kernel function chosen in defining the nonlinear mapping. Clearly, the choice of kernel is data specific; however, in the specific case of data partitioning, a kernel which will have universal approximation qualities such as RBF is most appropriate. In [6], we proposed kernel-based hybrid c-means clustering (KPFM) as an improvement over possibilistic fuzzy c-means clustering [8] using Gaussian kernel function. In most papers, Gaussian kernel function is used as the kernel function. Different kernels will induce different metric measures resulting in new clustering algorithms. Very few papers have studied other kernel functions such as hyper-tangent kernel function [7, 9]. In this paper, we have tried to investigate the effect of different kernel functions on the clustering results. To our knowledge, this is the first such comparison of kernel clustering algorithms using different kernel functions for general purpose clustering. The paper is organized as follows. A background of kernel-based approach is given in Sect. 2. Kernel-based hybrid c-means clustering with different kernel functions incorporated is described in Sect. 3. The experimental results and comparative analysis are given in Sect. 4 followed by the main conclusions presented in Sect. 5.

## 2 Kernel-Based Approach

A kernel function is a generalization of the distance metric that measures the distance between two data points as the data points are mapped into a high-dimensional space in which they are more clearly separable. By employing a mapping function,  $\Phi(x)$ , which defines a nonlinear transformation:  $x \rightarrow \Phi(x)$ , the nonlinearly inseparable data structure existing in the original data space can possibly be mapped into a linearly separable case in the higher-dimensional feature space. Given an unlabeled data set  $X = \{x_1, \dots, x_N\}$  in the  $p$ -dimensional space  $R^p$ , let  $\Phi$  be a nonlinear mapping function from this input space to a high-dimensional feature space  $H$ .

$$\Phi: R^p \rightarrow H, x \rightarrow \Phi(x)$$

It is possible to compute Euclidean distances in feature space without knowing explicitly  $\Phi$ . This can be done using kernel trick in which the computation of distances of vectors in feature space is just a function of the input vectors.

$$\begin{aligned} \|\Phi(x_k) - \Phi(v_i)\|^2 &= (\Phi(x_k) - \Phi(v_i)) \cdot (\Phi(x_k) - \Phi(v_i)) \\ &= \Phi(x_k) \cdot \Phi(x_k) - 2\Phi(x_k)\Phi(v_i) + \Phi(v_i) \cdot \Phi(v_i) \quad (1) \\ &= K(x_k, x_k) - 2K(x_k, v_i) + K(v_i, v_i) \end{aligned}$$

**Table 1** List of kernel functions

Name of kernel	Kernel function	Attribute
Gaussian	$\exp\left(-\ x_i - x_j\ ^2/2\sigma^2\right)$	$K(x, x) = 1$
Hyper-tangent	$1 - \tanh\left(\beta\ x_i - x_j\ ^2/2\sigma^2\right)$	$K(x, x) = 1$
Log	$\log\left(1 + \beta\ x_i - x_j\ ^2\right)$	$K(x, x) = 0$

Some examples of robust kernel functions are given in Table 1.

### 3 Kernel-Based Hybrid c-Means Clustering

We proposed a kernel-based hybrid c-means clustering (KPFCM) in [6] which used Gaussian kernel in the induced distance metric. The KPFCM model minimizes the following objective function:

$$J_{\text{KPFCM}}(U, V, T) = \sum_{k=1}^N \sum_{i=1}^c (au_{ik}^m + bt_{ik}^\eta) \|\Phi(x_k) - \Phi(v_i)\|^2 + \sum_{i=1}^c \gamma_i \sum_{k=1}^N (1 - t_{ik})^\eta \quad (2)$$

where  $\|\Phi(x_k) - \Phi(v_i)\|^2$  is the square of distance between  $\Phi(x_k)$  and  $\Phi(v_i)$ . Also  $0 \leq u_{ik} \leq 1$ ,  $t_{ik} < 1$ ,  $a > 0$ ,  $b > 0$ ,  $m > 1$  and  $\eta > 1$ . The constant  $a$  defines the relative importance of fuzzy membership, whereas  $b$  relates to the typicality value in the objective function. If we confine ourselves to the Gaussian kernel function which is used almost exclusively in the literature, then  $K(x, x) = 1$ .  $\|\Phi(x_k) - \Phi(v_i)\|^2 = 2(1 - K(x_k - v_i))$ . Thus, Eq. 2 can be rewritten as

$$J_{\text{KPFCM}}(U, V, T) = 2 \sum_{k=1}^N \sum_{i=1}^c (au_{ik}^m + bt_{ik}^\eta) (1 - K(x_k, v_i)) + \sum_{i=1}^c \gamma_i \sum_{k=1}^N (1 - t_{ik})^\eta \quad (3)$$

The update of  $u_{ik}$ ,  $v_i$  and  $t_{ik}$  is as follows:

$$u_{ik} = (1/(1 - K(x_k, v_i)))^{1/m-1} / \sum_{j=1}^c (1/(1 - K(x_k, v_j)))^{1/m-1} \quad (4)$$

$$v_i = \sum_{k=1}^N (au_{ik}^m + bt_{ik}^\eta) K(x_k, v_i) x_k / \sum_{k=1}^N (au_{ik}^m + bt_{ik}^\eta) K(x_k, v_i) \quad (5)$$

$$t_{ik} = 1 / \left(1 + [2b(1 - K(x_k, v_i))/\gamma_i]^{\frac{1}{\eta-1}}\right) \quad (6)$$

We now incorporate different kernel functions to examine and compare with Gaussian kernel. The names given to different clustering algorithms are KPFCM (using Gaussian kernel), KPFCM-H (hyper-tangent kernel), and KPFCM-L (log kernel). Using a hyper-tangent kernel function, the objective function for KPFCM-H clustering is as follows:

$$J_{\text{KPFCM-H}} = 2 \sum_{k=1}^N \sum_{i=1}^c (\text{au}_{ik}^m + \text{bt}_{ik}^\eta) \tanh\left(\frac{\|x_k - v_i\|^2}{\sigma^2}\right) + \sum_{i=1}^c \gamma_i \sum_{k=1}^N (1 - t_{ik})^\eta \quad (7)$$

The cluster center  $v_i$  for hyper-tangent kernel function is as follows:

$$v_i = \frac{\sum_{k=1}^N (\text{au}_{ik}^m + \text{bt}_{ik}^\eta) K(x_k, v_i) \left(1 + \tanh\left(\frac{\|x_k - v_i\|^2}{\sigma^2}\right) x_k\right)}{\sum_{k=1}^N (\text{au}_{ik}^m + \text{bt}_{ik}^\eta) K(x_k, v_i) \left(1 + \tanh\left(\frac{\|x_k - v_i\|^2}{\sigma^2}\right)\right)} \quad (8)$$

The expressions for  $u_{ik}$  and  $t_{ik}$  remain the same as in Eqs. 4 and 6, respectively. Using a log kernel function, the objective function for KPFCM-L clustering is as follows:

$$J_{\text{KPFCM-L}} = -2 \sum_{k=1}^N \sum_{i=1}^c (\text{au}_{ik}^m + \text{bt}_{ik}^\eta) \log\left(1 + \beta \|x_k - v_i\|^2\right) + \sum_{i=1}^c \gamma_i \sum_{k=1}^N (1 - t_{ik}) \quad (9)$$

The cluster center  $v_i$  for log kernel function is as follows:

$$v_i = \frac{\sum_{k=1}^N (\text{au}_{ik}^m + \text{bt}_{ik}^\eta) \left(1/1 + \beta \|x_k - v_i\|^2\right) x_k}{\sum_{k=1}^N (\text{au}_{ik}^m + \text{bt}_{ik}^\eta) \left(1/1 + \beta \|x_k - v_i\|^2\right)} \quad (10)$$

The update of  $u_{ik}$  and  $t_{ik}$  for log kernel function is as follows:

$$u_{ik} = \left(\log\left(1 + \beta \|x_k - v_i\|^2\right)\right)^{-1/(m-1)} / \sum_{i=1}^c \left(\log\left(1 + \beta \|x_k - v_i\|^2\right)\right)^{-1/(m-1)} \quad (11)$$

$$t_{ik} = 1 / \left(1 + (-b(\log(1 + \beta \|x_k - v_i\|^2) / \gamma_i))^{-1/(\eta-1)}\right) \quad (12)$$

## 4 Experimental Study

A series of experiments were run for a variety of data sets using the KPFCM clustering. The objective of this comprehensive suite of experiments is to come up with a thorough comparison of the performance of the KPFCM clustering using different kernel functions in the objective function. Many two-dimensional

**Table 2** Terminal centroids produced by KPFCM, KPFCM-H, and KPFCM-L on  $X_{12}$

Clustering	KPFCM	KPFCM-H	KPFCM-L
Centroids	[-3.33 0.002 3.33 0.002]	[-3.4 0 3.4 0]	[-3.33 0.001 3.33 0.001]
$E_*$	0.005	0	0.004

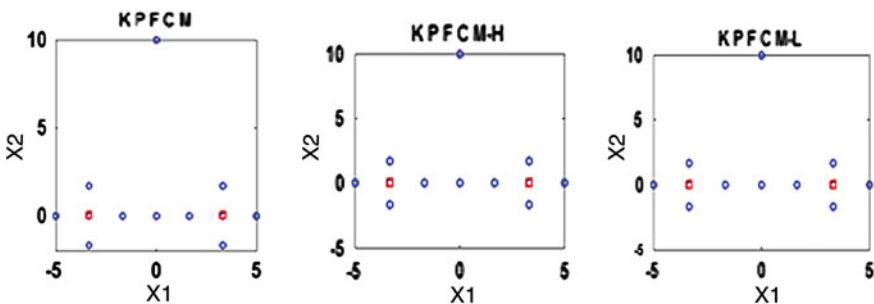
synthetic data sets were used with a wide variety in the shape of clusters, number of data points, and count of features of each datum. The real-life data sets used in the experiments are well-known Iris data sets.

### 4.1 Identical Data with Noise

The first simulation experiment involves the data set  $X_{12}$  [8]. The ideal (true) centroids for the  $X_{12}$  data set are  $V_{ideal} = \begin{bmatrix} -3.34 & 0 \\ 3.34 & 0 \end{bmatrix}$ . Let  $V_{KPFCM}^{12}$ ,  $V_{KPFCM-H}^{12}$ ,  $V_{KPFCM-L}^{12}$  be the final centroids identified by their respective algorithms. The results of the final centroids identified by their respective algorithms are tabulated in Table 2. To show the effectiveness of the proposed algorithm, we also compute the error  $E_* = \|V_{ideal} - V_*^{12}\|^2$ , where \* corresponds to KPFCM/KPFCM-H/KPFCM-L, respectively. Figure 1 displays the clustering results of three clustering algorithms with different kernel functions. The centroids produced by clustering techniques using different kernel functions are almost identical.

### 4.2 Dunn Data Set

In general, all clustering techniques perform well for pattern sets that contain patterns of similar volume and similar number of patterns. By changing the volume of clusters in a pattern set, we observe the effectiveness of different clustering



**Fig. 1** Clustering results for data  $X_{12}$

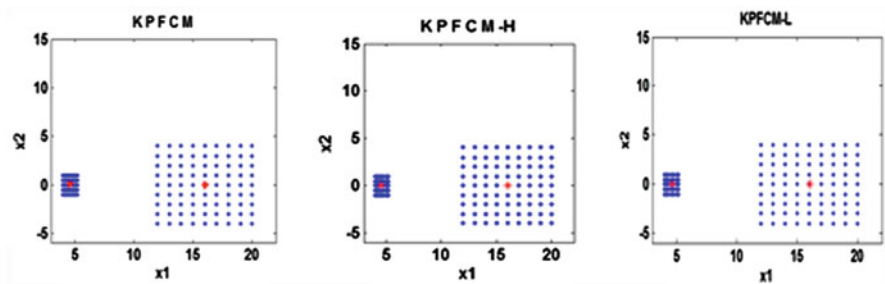


Fig. 2 Clustering results for Dunn data set

algorithms. Figure 2 displays the clustering results of three clustering algorithms. We see that all clustering algorithms correctly partition the two clusters with almost same accuracy.

### 4.3 Gaussian Random Data

In this example, a Gaussian random number generator was used to create a data set consisting of two clusters. The noise points are then added to the lower cluster to obtain a difference in volume compared to the upper cluster. Figure 3 shows the clustering results of three clustering algorithms. There are only two misclassifications in case of KPFCM-H. The results produced by KPFCM and KPFCM-L are completely identical.

### 4.4 Iris Data Set

This is a four-dimensional data set containing 50 samples each of three species of Iris flowers. One class is linearly separable from the other two; the latter are not linearly separable from each other. As indicated in Table 3, the typical result of

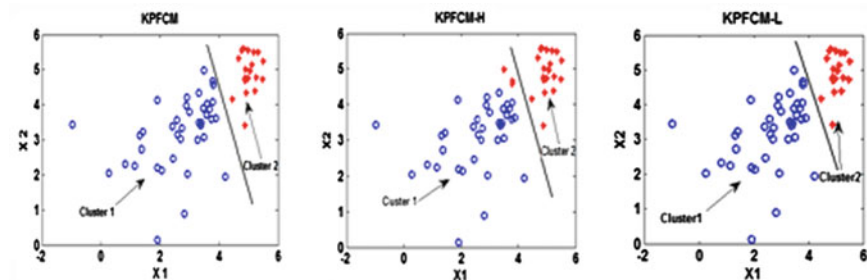


Fig. 3 Clustering results for Gaussian random data set

**Table 3** Number of misclassified data and accuracies using KPFCM, KPFCM-H, and KPFCM-L method for the Iris data set

Name of algorithm	Misclassifications	Accuracy (%)
KPFCM ( $\sigma = 0.5, a = 1, b = 2, \eta = 2$ )	11	92.6
KPFCM-H ( $\sigma = 0.25, a = 2, b = 2, \eta = 4$ )	11	92.6
KPFCM-L ( $\beta = 15, a = 1, b = 2, \eta = 2$ )	08	94.6

comparing KPFCM partitions to the physically correct labels of Iris is 11 errors. KPFCM-H also gives 11 errors. The number of misclassified data by KPFCM-L algorithm is 08 with an accuracy of 94.6 %.

## 5 Conclusions

This paper has presented the detailed analysis of different kernel functions on KPFCM clustering. In literature, most kernel-based clustering algorithms use Gaussian kernel functions. We have studied two nonGaussian kernel functions, namely hyper-tangent and log kernel functions. We incorporated these kernel functions in KPFCM clustering and applied these clustering algorithms on a wide variety of synthetic data as well as real-life data set. From the experiments, we have seen that the two nonGaussian kernel functions have worked as well as Gaussian kernel function. To summarize, we conclude that we have another class of robust kernel functions that have worked well in typical clustering examples of nonlinear classification boundaries.

## References

1. K.R. Muller et al., An Introduction to Kernel-based Learning Algorithms, IEEE Trans. on Neural Networks, 12 (2) (2001) 181-202.
2. M. Girolami, Mercer Kernel-based Clustering in Feature Space, IEEE Trans. on Neural Networks, 13 (3) (2002) 780-784.
3. F. Camastra, A. Verri, A Novel Kernel Method for Clustering, IEEE Transaction on Pattern Analysis and Machine Intelligence, 27 (5) (2005), 801-805.
4. Z.d.Wu, W.X.Xie, Fuzzy c-means Clustering Algorithm Based on Kernel Methods in: Proc. Of fifth Intl. Conf. on Computational Intelligence and Multimedia Applications (2003) 47-54.
5. D.Q. Zhang, S.C. Chen, Fuzzy Clustering using Kernel Methods, in: Proc. of Intl. Conf. on Control and Automation, China, (2002) 123-128.
6. M. Tushir, S. Srivastava, A New Kernelized Hybrid c-means Clustering Model with Optimized Parameters, J. Applied Soft computing, 10 (2) (2010) 381-389.
7. D.Q. Zhang, S.C. Chen, Clustering Incomplete Data using Kernel-based Fuzzy c-means Algorithm, Neural Processing Letters, 18 (3) (2003) 155-162.
8. N.R. Pal, K. Pal, J. Keller, J.C. Bezdek, A Possibilistic Fuzzy c-means Clustering Algorithm, IEEE Trans. Fuzzy Syst., 13 (4) (2005) 517-530.
9. S. R. Kannan et al, Robust Kernel FCM in Segmentation of Breast Medical Images, Intl Journal Expert Systems with Applications, 38 (4) (2011), 4382-4389.