Sādhanā Vol. 38, Part 3, June 2013, pp. 407–419. © Indian Academy of Sciences

Single pass kernel *k*-means clustering method

T HITENDRA SARMA[1,*], P VISWANATH[2]
and B ESWARA REDDY[3]

[1]Department of Computer Science and Engineering, Srinivasa Ramanujan Institute of Technology, Anantapur 515701, India
[2]Department of Computer Science and Engineering, Rajeev Gandhi Memorial College of Engineering and Technology, Nandyal 518501, India
[3]Department of Computer Science and Engineering, Jawaharlal Nehru Technological University, Anantapur College of Engineering, Anantapur 515002, India
e-mail: hitendrasarma@ieee.org; viswanath.p@ieee.org; eswarcsejntu@gmail.com

MS received 30 April 2012; revised 2 January 2013; accepted 13 February 2013

Abstract. In unsupervised classification, kernel *k*-means clustering method has been shown to perform better than conventional *k*-means clustering method in identifying non-isotropic clusters in a data set. The space and time requirements of this method are $O(n^2)$, where n is the data set size. Because of this quadratic time complexity, the kernel *k*-means method is not applicable to work with large data sets. The paper proposes a simple and faster version of the kernel *k*-means clustering method, called *single pass kernel k-means clustering method*. The proposed method works as follows. First, a random sample $\mathcal{S}$ is selected from the data set $\mathcal{D}$. A partition $\Pi_{\mathcal{S}}$ is obtained by applying the conventional kernel *k*-means method on the random sample $\mathcal{S}$. The novelty of the paper is, for each cluster in $\Pi_{\mathcal{S}}$, *the exact cluster center* in the input space is obtained using the *gradient descent approach*. Finally, each unsampled pattern is assigned to its closest exact cluster center to get a partition of the entire data set. The proposed method needs to scan the data set only once and it is much faster than the conventional kernel *k*-means method. The time complexity of this method is $O(s^2+t+nk)$ where s is the size of the random sample $\mathcal{S}$, k is the number of clusters required, and t is the time taken by the gradient descent method (to find exact cluster centers). The space complexity of the method is $O(s^2)$. The proposed method can be easily implemented and is suitable for large data sets, like those in data mining applications. Experimental results show that, with a small loss of quality, the proposed method can significantly reduce the time taken than the conventional kernel *k*-means clustering method. The proposed method is also compared with other recent similar methods.

Keywords. Data mining; unsupervised classification; kernel *k*-means clustering method; gradient descent method.

*For correspondence

1. Introduction

Data clustering is a process of identifying the natural groupings that exists in a given data set, such that the objects in the same cluster are more similar and the objects in different clusters are less similar (in other words, dissimilar). It has been considered as an important tool in various applications like pattern recognition, image processing, data mining, remote sensing, statistics, etc. (Jain *et al* 1999). Clusters in the given data may be of different types, such as, isotropic, non-isotropic, linearly separable, non-linearly separable, etc. It has been observed that, when, data sets have isotropic and linearly separable clusters in the input space, sum-of-squares based partitioning methods, like conventional *k-means clustering method*, are effective. The *k-means method* find k patterns, each represents one cluster, by solving an optimization problem using gradient descent procedure in linear time (Bottou & Bengio 1995; Hitendra Sarma & Viswanath 2009). On the other hand, kernel-based clustering methods, like *kernel k-means clustering method*, are proved to be effective in identifying clusters that are non-isotropic and which are linearly inseparable in the input space (Müller *et al* 2001; Girolami 2002).

Kernel k-means clustering method is a non-linear extension of the conventional k-means clustering method (Dhillon *et al* 2005). Girolami (2002) first proposed the *kernel k-means clustering method*. It is an iterative method. It first maps the data points from the input space to a higher dimensional feature space through a nonlinear transformation $\phi(\cdot)$ and then minimizes the clustering error in that feature space. The distance between a data point and a cluster center in the feature space (i.e., the kernel-induced feature space) can be computed using a *kernel function* without knowing the explicit form of the transformation (Scholkopf *et al* 1998). This is, because, the dot product between the images of two data points X and Y in the feature space, which is $\phi(X) \cdot \phi(Y)$, can be computed as a function $K(X, Y)$, where $K : D \times D \rightarrow \mathbb{R}$ is called the *kernel function*. This is often known as the kernel trick and is valid for transformations that satisfies Mercer's conditions (Critianini & Shewe-Taylor 2000). Some standard kernel functions are given below.

- Polynomial kernel of degree $p : K(X, Y) = (X \cdot Y + c)^p$, where p is a positive integer and $c \in \mathbb{R}$.
- Gaussian (RBF) kernel: $K(X, Y) = exp(-\frac{||X-Y||^2}{2\sigma^2})$, where $\sigma \in \mathbb{R}$ is called Gaussian kernel parameter.
- Sigmoidal kernel : $K(X, Y) = \tanh(a(X \cdot Y) + b)$, where $a, b \in \mathbb{R}$

For two arbitrary data points X_i and X_j, very often, in the iterative process of clustering, $K(X_i, X_j)$ is needed. So, a matrix called kernel matrix $H = [K_{ij}]_{n \times n}$ is found, where the $(i, j)^{th}$ entry is $K_{ij} = K(X_i, X_j)$. Here n is the data set size. The kernel matrix is precomputed and stored. So, the time and space requirements (which is given, in detail, in later sections) are $O(n^2)$. This is the drawback of kernel k-means clustering method and because of which it is not suitable when the data set size is large. Several other drawbacks of the kernel k-means method includes, the need of prior knowledge about the number of clusters(k), dependency of the clustering result on the seed points as well as on the kernel function (Kim *et al* 2005; Radha Chitta *et al* 2011).

Several improvements are proposed to cater these drawbacks. In order to overcome the local minima problem, Tzortzis & Likas (2008) proposed the *global kernel k-means method*, which produces a final partition which is independent of the initial seed points. *Soft geodesic kernel k-means method* (Kim *et al* 2007) improves the quality of the clustering result by taking the internal data manifold structure into account. Further, some semi-supervised clustering algorithms were aimed to improve the clustering accuracy under the supervision of a limited amount

of labelled data. Kernel based approaches, such as, kernel based c-means method (Wu & Xie 2003),[1] kernel-based fuzzy c-means method, semi-supervised kernel fuzzy c-means method (Zhang & Lu 2009), etc., have been successfully used to deal with classification and clustering problems. Spectral clustering methods are also used to identify nonlinearly separable clusters in the input space. Some formal arguments show that, both kernel and spectral methods have some similarities (Dhillon *et al* 2004, 2005). Some other related methods are reviewed in (Filippone *et al* 2008).

The main focus of the present paper is to reduce the time and space complexities of the kernel k-means clustering method. To the best of our knowledge there are only few improvements proposed to handle the time and space complexity problems of the kernel k-means method. Zhang & Rudnicky (2002) have proposed a new block based scheme which addressed the space complexity of kernel k-means method in case of large data sets. However, this method also requires to compute the full kernel matrix. Recently, Radha Chitta *et al* (2011) have presented a sample-based approach called the *two-step kernel k-means method*. This sample-based approach results in a great reduction in the running time, but yields only approximate clustering results. Further, Radha Chitta *et al* (2011) have also proposed a randomized approach called *the approximate kernel k-means clustering method* which has reduced time and space requirements and which produces a similar clustering result as the conventional kernel k-means method. More recently, Hitendra Sarma *et al* (2011) have proposed a prototype-based hybrid approach which gives a similar clustering result as the conventional kernel k-means method but in much reduced time. These methods are discussed in detail in the subsequent sections.

This paper presents a simple and faster version of the kernel k-means method, called the *single pass kernel k-means* clustering method. The proposed method works as follows. First, a random sample $\mathcal{S}$ is selected from the data set $\mathcal{D}$ and applies the conventional kernel k-means method on $\mathcal{S}$ to derive a partition of it, say $\Pi_{\mathcal{S}}$. Then the *exact cluster center* for each cluster in $\Pi_{\mathcal{S}}$ is found using the gradient descent method (Radha Chitta *et al* 2011). Finally, each unsampled pattern is assigned to its closest cluster center in the induced space to get a partition of the entire data set $\Pi_{\mathcal{D}}$.

In the proposed method, the key point is to find *the exact center* of a cluster which is in the kernel-induced feature space. Note, since the transformation function is unknown, it is not possible to obtain cluster centers in the feature space. But, a point X in the input space can be found such that $\phi(X)$ is the exact cluster center (for the corresponding cluster in the feature space). This is done by applying the gradient descent procedure (Bottou & Bengio 1995; Duda *et al* 2000) where the objective reduced is the one in the feature space, but the solution obtained is a representative pattern in the input space (explained in detail in section 4). In the existing literature, often, a pseudo center is used instead of the exact center (Zhang & Rudnicky 2002; Radha Chitta *et al* 2011). Pseudo center of a cluster C is a pattern in the input space which is nearest to the centroid of the patterns that are grouped into the cluster C. Note that, the pseudo center of C, which is a pattern in the input space, may not represent the exact cluster center of C in the induced-feature space. The proposed method needs to scan the data set only once and it is much faster than the conventional kernel k-means method and hence it is applicable for large data sets.

The paper is organized as follows. Section 2 briefly reviews the kernel k-means clustering method, while section 3 outlines the related work carried out in the literature to reduce the time and space complexities of the kernel k-means clustering method. The proposed method, called

[1] Some authors call the k-means clustering method as the c-means clustering method.

the *single pass kernel k-means* clustering method, is described in section 4. Experimental results are given in section 5 and section 6 gives some of the conclusions and future work.

2. kernel *k*-means clustering method

Let $\mathcal{D} = \{X_1, X_2, \ldots, X_n\}$ be the data set of size n. Let k be the number of clusters required and $\mathcal{M}^{(0)}$ be the initial seed points. Note, $\mathcal{M}^{(0)} = \{m_1^{(0)}, m_2^{(0)}, \ldots, m_k^{(0)}\}$ are points in the input space. For each pattern $X \in \mathcal{D}$, assign X to the mean $m_i^{(0)}$, such that $m_i^{(0)}$ is at the nearest distance form X in the kernel-induced feature space. Let the resultant initial partition be $\Pi^{(0)}$. The kernel k-means method takes $\mathcal{D}$, k and $\Pi^{(0)}$ as input parameters and produces a partition of the entire data set, $\Pi_{\mathcal{D}}$ as the output.

The objective function is to minimize the criterion function

$$J = \sum_{j=1}^{k} \sum_{X_i \in C_j} ||\phi(X_i) - \mu_j||^2, \tag{1}$$

where μ_j is the mean of cluster C_j. That is

$$\mu_j = \sum\nolimits_{X_i \in C_j} \frac{\phi(X_i)}{|C_j|} \tag{2}$$

in the induced space.

Distance between two data points $\phi(X_i)$ and $\phi(X_j)$ in the induced space, that is

$$\begin{aligned} &||\phi(X_i) - \phi(X_j)||^2 \\ &\quad = \phi^2(X_i) - 2\phi(X_i) \cdot \phi(X_j) + \phi^2(X_j) \\ &\quad = K(X_i, X_i) - 2K(X_i, X_j) + K(X_j, X_j). \end{aligned} \tag{3}$$

Further, $||\phi(x_i) - \mu_j||^2$ can be calculated without knowing the transformation $\phi(\cdot)$ explicitly as given below:

$$\begin{aligned} &||\phi(X_i) - \mu_j||^2 \\ &\quad = ||\phi(X_i) - \sum\nolimits_{X_l \in C_j} \frac{\phi(X_l)}{|C_j|}||^2 \\ &\quad = \phi(X_i) \cdot \phi(X_i) - \mathrm{F}(X_i, C_j) + \mathrm{G}(C_j), \end{aligned} \tag{4}$$

where

$$\mathrm{F}(X_i, C_j) = -\frac{2}{|C_j|} \sum\nolimits_{X_l \in C_j} \phi(X_l) \cdot \phi(X_i) \tag{5}$$

and

$$\mathrm{G}(C_j) = \frac{1}{|C_j|^2} \sum_{X_l \in C_j} \sum_{X_s \in C_j} \phi(X_l) \cdot \phi(X_s) \tag{6}$$

The iterative method is outlined in the algorithm 1. The time complexity and the space complexity of the kernel k-means clustering method are $O(n^2)$.

Algorithm 1. kernel-k-means-clustering-method($\mathcal{D}$, k, $\Pi^{(0)}$).

1. For each cluster C_j, find $|C_j|$ and $G(C_j)$.
2. Compute $F(X_i, C_j)$ for each X_i and for each cluster C_j.
3. Find $||\phi(X_i) - \mu_j||^2$ using equation (4) and assign X_i to its nearest center.
4. Update μ_j, for $j = 1$ to k, using equation (2).
5. Repeat step 1 through step 4 till convergence.
Output:
The final partition $\Pi_{\mathcal{D}} = \{C_1, C_2, \ldots, C_k\}$.

3. Related work

This section outlines the recent improvements proposed to address the running time and space requirement problems of the kernel k-means clustering method for large data sets.

Zhang & Rudnicky (2002) proposed a block-based approach to address the space complexity of the method for large data sets. In this approach, the kernel matrix H is pre-computed and stored in the secondary memory before starting the iterative process. That is the size of H can be theoretically extended to as large as the entire disk. Later, the kernel matrix H is split into blocks, where size of each block is determined according to the I/O capability and the available main memory size. In each iteration, each block is moved as a whole from secondary memory to the main memory and processed. So the number of I/O operations for each iteration is equal to the number of blocks, but it is a costly operation when there are more number of blocks. Although this block-based approach handles the space complexity, it still requires the computation of the full kernel matrix. Hence this approach is also not a good choice for large data sets as the time requirement is not reduced.

Radha Chitta *et al* (2011) have proposed a randomized approach called the *approximate kernel k-means* to reduce the total running time and space requirements of the conventional kernel k-means clustering method. In this method a random sample, say B, which is of size q is selected and the kernel similarity matrix, called H_B, between the data points in $\mathcal{D}$ and the sampled q data points is calculated. The size of the matrix H_B is $n \times q$ which is very less when compared to the size of the full kernel matrix H which is of size $n \times n$, for $q << n$. The iterative process starts with an initial partition. In each iteration, the matrix H_B is used to estimate the closest cluster center for each data point. The process iterates till there is no change in the cluster members of each cluster. The time complexity of this method is $O(q^2kr + qnkr + q^2n)$, where r is the number of iterations till convergence. The space complexity of this method is $O(nq)$.

Recently, Hitendra Sarma *et al* (2011) have proposed a prototype-based hybrid approach to address both time and space complexities of the kernel k-means clustering method. This runs in two stages. In the first stage, the data set is divided in to a number of grouplets by employing a modified version of the leaders clustering method (Spath 1980), called the *kernel based leaders clustering method* (Hitendra Sarma *et al* 2011) such that the grouplets are found in the kernel space (not in the input space), but each grouplet is represented by a prototype, called leader, which is pattern in the input space. The set of leaders, which depends on a threshold parameter, can be derived in $O(n)$ time. In the second stage, kernel k-means clustering method is applied with the set of leaders to derive a partition of the set of leaders. Finally, each leader is replaced by its group to get a partition of the data set. The time complexity of this hybrid method is $O(n + p^2)$, where p is the leaders set size. Its space complexity is also $O(n + p^2)$. For

appropriately selected threshold, this method runs in a faster pace than the conventional kernel k-means method with a small deviation in the clustering result. However, the optimal threshold value depends on the data set and it is an additional overhead to find the optimal threshold before hand.

A simple and naive approach, called the *two-step kernel k-means method*, is presented in Radha Chitta *et al* (2011) for reducing the computational complexity of the kernel k-means method. In the first step, a random sample of s data points is selected from the data set, and the optimal cluster centers only based on the sampled data points are found. Later, in the second step every unsampled data point is assigned to the cluster whose center is the nearest in the induced space. Here, the pseudo cluster centers are used to represent the optimal cluster centers in the induced space. This approach has reduced both time complexity and memory requirements. However, the clustering result of this method will be very much deviated form that obtained using the conventional kernel k-means method. This is because of the fact that pseudo cluster centers in the input space may not represent the exact cluster centers in the kernel induced feature space. The deviation in the clustering result can be reduced by using the exact cluster centers instead of the optimal (pseudo) cluster centers in the induced space to generate the final partition of the data set. This is the motivating step of the proposed method which is explained in the following section.

4. Single pass kernel k-means clustering method

This section describes the proposed *single pass kernel k-means* clustering method. Let $\mathcal{S} = \{X_1, X_2, \ldots, X_s\}$ be a random sample of the data set $\mathcal{D}$. Here s denotes the number of patterns in the random sample $\mathcal{S}$. Let $\mathcal{M}^{(0)}$ be the set of initial seed points and $\Pi^{(0)}$ be the initial partition of the data set. The kernel k-means clustering method is applied over the random sample $\mathcal{S}$ to derive a partition of $\mathcal{S}$ and it is denoted by $\Pi_{\mathcal{S}}$. Let $\Pi_{\mathcal{S}} = \{C_1^{\mathcal{S}}, C_2^{\mathcal{S}}, \ldots C_k^{\mathcal{S}}\}$.

For each cluster $C_j^{\mathcal{S}}$ in $\Pi_{\mathcal{S}}$, the exact center is a pattern X which is in the input space such that the criterion function $J(X) = \sum_{X_i \in C_j^{\mathcal{S}}} ||\phi(X_i) - \phi(X)||^2$ is minimized, where $X = (x_1, x_2, \ldots, x_d)^T$ is a d-dimensional vector in the input space. The gradient descent method is applied to find the pattern X^* such that the objective function is minimum at $X = X^*$. Since the objective function is a convex function, there is no local minima problem. The pattern X^* is taken as the exact center of the cluster $C_j^{\mathcal{S}}$ and is denoted by M_j.

Gradient descent method is an iterative approach which starts with the initial guess of the exact cluster center and in each iteration it is updated by searching along the direction of the negative gradient of the criterion function J. The procedure to find the exact cluster center for each cluster is given in algorithm 2.

Finally, assign each unsampled pattern to its closest exact cluster center in the induced space to get a partition of the entire data set $\Pi_{\mathcal{D}}$.

The proposed method needs to scan the data set only once. The overall running time of the proposed method depends on the size of the initial random sample $\mathcal{S}$. The time complexity of this method is $O(s^2 + t + nk)$, where s is the size of the random sample $\mathcal{S}$, t is the time required to find the k exact cluster centers using the gradient descent approach, n is the size of the entire data set and k is the number of clusters required. Where as the space complexity of the method is $O(s^2)$. Since $s << n$, the proposed method is much faster than the conventional kernel k-means method. Further, the space requirement is also reduced. The proposed method can be easily implemented and it is applicable to work with large data sets. The proposed method is given in algorithm 3.

Algorithm 2. Gradient-descent-approach ($\Pi_{\mathcal{S}}$).

for each cluster $C_j^{\mathcal{S}} \in \Pi_{\mathcal{S}}$ **do**
  Let the initial guess for exact cluster center of $C_j^{\mathcal{S}}$ be $M_j^{(0)}$.
  Let $r = 0$; /* r is the iteration number */
  repeat
    Find $\nabla_X(J)$
    Find $M_j^{(r+1)}$ such that $M_j^{(r+1)} = M_j^{(r)} - \alpha \nabla_X J(M_j^{(r)})$
    /* where α is called the learning rate */
    Put $r = r + 1$;
  until $(M_j^{(r)} == M_j^{(r-1)})$
  The exact center of the cluster $C_j^{\mathcal{S}}$ *i.e.,* $M_j = M_j^{(r)}$
end for
Output: The set of exact centers $\{M_1, M_2, \ldots M_k\}$.

Algorithm 3. Single pass kernel k-means clustering method ($\mathcal{D}$, k, $\mathcal{M}^{(0)}$).

1. Select a random sample $\mathcal{S}$ from the data set $\mathcal{D}$. Let p be the size of the random sample $\mathcal{S}$.
2. Apply kernel-k-means-clustering-method($\mathcal{S}$,k,$\Pi^{(0)}$). Let $\Pi_{\mathcal{S}}$ be the output.
3. Apply Gradient-descent-approach ($\Pi_{\mathcal{S}}$) to get the exact cluster centers for the k clusters in the partition $\Pi_{\mathcal{S}}$. Let the set of exact cluster centers be $\{M_1, M_2, \ldots M_k\}$.
4. Assign each pattern $X_i \in \mathcal{D}$ to the cluster C_j such that M_j is at the nearest distance from X_i in the induced feature space. Let the resultant partition is $\Pi_{\mathcal{D}}$.
5. Output $\Pi_{\mathcal{D}}$.

5. Experimental study

In this section, we extensively study the performance of the proposed method over various bench mark data sets as well as some synthetic data sets. The proposed method is compared with the conventional kernel k-means method and it is also compared with the recent similar methods viz., two-step kernel k-means method (Radha Chitta *et al* 2011), approximate kernel k-means method (Radha Chitta *et al* 2011) and the prototype-based hybrid kernel k-means clustering method (Hitendra Sarma *et al* 2011). The comparison is done both in terms of Clustering Accuracy(CA) and the overall Running Time(RT). CA of a clustering method is the percentage of similarity between the two partitions, one is the actual partition of the data set and the other is the partition obtained using the corresponding clustering method, where the similarity is found using *Rand-Index* (Rand 1971; Hubert & Arabie 1985). The RBF kernel is used in all these methods. The experiments were conducted on a PC with an *intel P4 processor (3.2 Ghz)* with 512 MB RAM.

5.1 *Data sets*

The data sets employed in this empirical study include: Banana, Rings, Concentric Circles, Desert, Iris, Handwritten Symbols, Pendigits, Optical Character Recognition (OCR), Letter Image Recognition (LIR), Shuttle and Gaussian data sets. Iris and Pendigits data sets are available at the UCI machine learning repository (Murphy 1994). Letter Image Recognition (LIR),

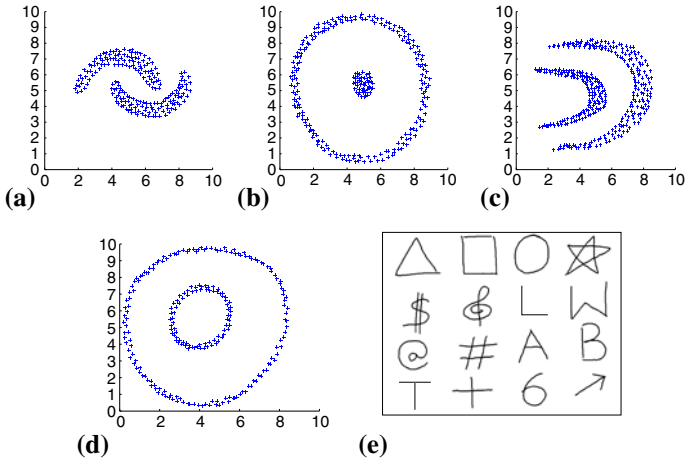

Figure 1. The artificial data sets used in the experiments. (**a**) Banana data set, (**b**) Rings data set, (**c**) Desert data set, (**d**) Concentric Circles data set and (**e**) Handwritten symbols data set.

Shuttle and OCR data sets are also used in Ravindra Babu & Narasimha Murty (2001) and Hitendra Sarma *et al* (2011, 2012). Banana, Rings, Concentric Circles, Desert, Handwritten symbols and Gaussian data sets are artificially generated.

The synthetic data sets used in our experiments include various types of clusters (see figure 1).

For Handwritten Symbols data set, sixteen hand-drawn symbols, as shown in figure 1(e) are used. Ten different persons are asked to draw each symbol for ten times. So, a total of 1600 hand-drawn symbols database is created. Each symbol's coordinates are collected by drawing it on a 512×512 writing pad. Zernike moments (Flusser *et al* 2009) of order 8 are used to represent each symbol after doing normalization against scale and translation as described in Hse & Richard Newton (2004).

The Gaussian data set, which is size of 60,000 and dimensionality 2, is generated from a tri-modal gaussian distribution $p(x) = \frac{1}{3}N(\mu_1, \Sigma_1) + \frac{1}{3}N(\mu_2, \Sigma_2) + \frac{1}{3}N(\mu_3, \Sigma_3)$. For dimensionality 2, $\mu_1 = (0, 0)$, $\mu_2 = (5, 5)$, $\mu_3 = (-5, 5)$ and the covariance matrix is taken as the Identity matrix of size $d \times d$, where d is the dimensionality of the data.

All the data sets have only numeric valued features. The properties of all the data sets are given in table 1.

5.2 *Analysing the efficiency of the proposed method*

This section analyses the efficiency of the proposed method. Further, the proposed method is also compared with the other existing similar methods proposed in Hitendra Sarma *et al* (2011) and Radha Chitta *et al* (2011).

The clustering accuracy(CA) and the total running time(RT) of the proposed method depends on the size of the initial random sample. To analyse this, the proposed method is executed on each data set in table 1 by varying the size of the initial random sample. For each size of the random sample, the proposed method is executed for 10 times by varying the sampled patterns.

The efficiency of the prototype-based hybrid method (Hitendra Sarma *et al* 2011) depends on the number of leaders p which inturn depends on the threshold. For threshold $= 0$, we have $p = |\mathcal{D}|$, with an additional computational overhead of $O(n)$ which is the computation cost of

Table 1. Details of both synthetic and benchmark data sets used.

Dataset	Number of patterns(n)	Number of dimensions(d)	Number of clusters(k)
Banana	200	2	2
Rings	250	2	2
Concentric circles	230	2	2
Desert	350	2	2
Iris	150	3	4
Handwritten symbols	1600	8	16
Pendigits	10992	16	10
OCR	10003	192	10
LIR	20000	16	26
Shuttle	58000	9	7
Gaussian	60000	2	3

deriving the leaders set, wherein each pattern in $\mathcal{D}$ becomes a leader (assuming, no duplicates in the data). Increasing the threshold value can decrease total running time but the clustering accuracy may also be decreased.

The criterion to find the optimal value for the threshold can be posed as the following optimization problem. Find the optimal threshold which maximizes the criterion

$$\{\textit{Clustering Accuracy} \; - \; \lambda \; \cdot \; \textit{Running Time}\} \, . \tag{7}$$

Here λ is a trade-off parameter between the clustering accuracy and the total running time (viz., increasing the clustering accuracy and reducing the running time) which has to be fixed based on availability of computational resources and on the application domain. In this experimental study, the threshold is estimated by taking $\lambda = 0.5$. Further, the set of leaders depends on the order of scanning of the data set (Hitendra Sarma *et al* 2011). Hence the results of this method are averaged over 10 runs by varying the order of scanning of the data set.

The efficiency of two-step kernel k-means and approximate kernel k-means clustering methods (Radha Chitta *et al* 2011) depends on the size of the initial random sample. For the sake of comparative study, the size of the random sample is taken to be the same as that of the number of

Table 2. Clustering accuracy(CA) obtained using various clustering methods for different sample sizes of LIR data set.

	Clustering accuracy (CA)			
Sample size	Two-step kernel k-means	Approximate kernel k-means	Prototype-based method	Single pass kernel k-means
12500	69.41±1.1	89.02±0.9	89.96±1.4	78.91±1.5
10000	58.2±1.4	81.5±1.6	84.2±1.5	69.2±2.1
5000	53.5±1.7	79.5±1.8	81.5±1.6	61.48±2.5
1000	48.9±2.6	75.3±2.4	80.3±1.9	58.5±2.8
500	42.8±2.7	68.9±3.1	75.9±2.1	55.9±2.9
100	35.1±2.9	64.6±3.2	69.2±2.8	52.1±3.3
50	31.8±3.6	60.4±3.7	65.1±3.5	50.9±3.6

Table 3. Running time (RT) of various clustering methods for different sample sizes of LIR data set.

	Running time (RT)			
Sample size	Two-step kernel k-means	Approximate kernel k-means	Prototype-based method	Single pass kernel k-means
12500	56.45±8.1	359.43±74.1	172.91±11.9	120.1±15.9
10000	32.14±1.4	142.3±25.14	51.47±4.2	36.59±12.2
5000	10.6±1.8	88.79±12.1	42.42±2.3	30.88±2.1
1000	2.54±0.5	54.3±5.6	32.6±2.1	20.56±1.3
500	0.24±0.02	38.1±3.8	9.5±0.7	3.9±0.8
100	0.09±0.002	24.5±2.9	0.9±0.02	0.6±0.07
50	0.005±0.001	15±1.2	0.2±0.03	0.1±0.02

leaders (p) used in the prototype-based method. The results of these two methods are averaged over 10 runs (each run with a different random sample).

The experimental results on the LIR data set are presented in tables 2 and 3. Similar type of results can be obtained on the other data sets also. The results presented in tables 2 and 3 show that the proposed method is much faster than the other existing similar methods with a small reduction in the clustering accuracy.

Finally, the proposed method is compared with the conventional kernel k-means method along with the above mentioned approximate methods for various data sets that are given in table 1. The efficiency of the conventional kernel k-means method depends on the initial seed points. So the conventional method is executed for 10 time by varying the initial seed points and the average CA and RT values for each data set are recorded. In case of the prototype-based hybrid method (Hitendra Sarma *et al* 2011), first we compute the best threshold value for each data

Table 4. Clustering accuracy(CA) obtained using the kernel k-means method, two-step kernel k-means method, approximate kernel k-means method, Prototype-based hybrid kernel k-means method and the proposed single pass kernel k-means clustering methods for various data sets.

	Clustering accuracy (CA)				
Data set	Kernel k-means method	Two-step kernel k-means	Approximate kernel k-means	Prototype-based method	Single pass kernel k-means
Banana	90.56±2.3	55.45±4.24	88.4±2.85	89.22±3.1	78.84±1.8
Rings	95.1±1.6	48.91±3.28	89.3±1.97	91.45±2.5	81.24±2.1
Concentric circles	74.4±5.9	41.40±6.1	71.1±4.2	73.6±3.1	63.15±1.2
Desert	74.88±3.8	48.56±5.8	71.454±3.2	72.39±2.6	66.21±2.5
Iris	89.76±4.3	59.02±6.4	78.43±2.5	88.59±1.5	69.26±1.5
Handwritten symbols	85.2±5.9	62.15±8.9	83.49±3.14	84.6±2.1	72.17±1.8
Pendigits	92.78±2.6	53.50±3.1	89.8±2.1	89.56±1.2	81.26±0.9
OCR	86.38±1.9	64.12±9.2	81.90±1.3	84.00±2.8	78.14±2.1
LIR	92.92±2.7	69.41±6.1	87.02±4.2	89.96±1.7	81.2±2.8
Shuttle	93.12±1.6	70.32±7.5	89.3±1.9	90.14±1.8	82.14±1.8
Gaussian	98.32±0.9	62.2±5.2	91.78±2.01	94.33±2.5	86.33±1.5

Table 5. Running times (RT) of the kernel k-means method, two-step kernel k-means method, approximate kernel k-means method, Prototype-based hybrid kernel k-means method and the proposed single pass kernel k-means clustering methods for various data sets.

	Running time (RT)				
Data set	Kernel k-means method	Two-step kernel k-means	Approximate kernel k-means	Prototype-based method	Single pass kernel k-means
Banana	0.022±0.001	0.002±0.001	0.006±0.002	0.004±0.001	0.003±0.001
Rings	0.054±0.002	0.002±0.001	0.016±0.001	0.009±0.001	0.005±0.001
Concentric circles	0.029±0.001	0.003±0.002	0.008±0.001	0.006±0.001	0.004±0.001
Desert	0.096±0.003	0.008±0.006	0.028±0.001	0.017±0.012	0.012±0.003
Iris	0.019±0.001	0.002±0.001	0.006±0.001	0.004±0.001	0.003±0.002
Handwritten symbols	36.21±2.54	3.52±2.3	10.30±3.2	6.25±1.64	4.1±1.2
Pendigits	418.078±56.21	18.34±2.1	84.01±9.1	34.123±2.86	26.12±1.61
OCR	841.5±98.33	23.65±4.5	132.12±12.5	34.777±6.27	29.34±3.6
LIR	3261.337±421.10	56.45±12.1	359.431±74.12	172.901±15.98	81.35±15.8
Shuttle	15780.25±544.98	154.55±41.2	4012.21±542.01	2280±129.21	1231±57.21
Gaussian	20040.0±1172.5	101.68±34.9	2972.11±620.54	1920±108.79	1043±57.21

set by following the criterion (7). For each set of initial seed points that are used in the conventional method, the prototype-based method is executed for 10 times by varying the order of scanning of the data set. The averaged CA and RT values are recorded. In case of two-step kernel k-means, approximate kernel k-means clustering method and the proposed method the size of the initial random sample is fixed to be the same as the size of the set of leaders in the prototype-based method. The results of all these methods are averaged over 10 runs by varying the initial sampled patterns. The averaged CA and RT values of all the methods are recorded for each data set. Tables 4 and 5 present the overall CA and RT of all the methods for various data sets, respectively.

6. Conclusions and future work

The paper presented a single pass kernel k-means method to address the time and space complexity problems of the conventional kernel k-means clustering method. The method first finds k exact cluster centers using only a few selected sample patterns form the data set. Later each unsampled pattern in the data set is assigned to its nearest exact cluster center to derive a partition of the entire data set. The proposed method requires to scan the data set only once and it is easy to implement. Hence it is applicable for large data sets. Further, the time and space complexities are much reduced when compared to the conventional kernel k-means method.

The proposed method is also compared with other recent similar techniques which include: (i) the two-step kernel k-means method (Radha Chitta *et al* 2011), (ii) the approximate kernel k-means method (Radha Chitta *et al* 2011) and (iii) Prototype-based hybrid kernel k-means method (Hitendra Sarma *et al* 2011). According to the experimental study, the proposed method achieved a great reduction in running time with a small reduction in the clustering accuracy when

compared to the conventional kernel k-means, approximate kernel k-means and the prototype-based methods, particularly for large data sets. The two-step kernel k-means method is much faster than the proposed method, but with a more loss of clustering accuracy. Future work is to further improve the accuracy of the proposed method.

Acknowledgements

The work reported in this paper is supported by a sponsored project from 'All India Council for Technical Education, New Delhi' under Research Promotion Scheme (RPS) scheme with reference: 'F.No: 8023/BOR/RID/RPS-51/2009–10'.

References

Critianini N and Shewe-Taylor J 2000 *Introduction to Support Vector Machines: And Other Kernel-Based Learning Methods*. Cambridge, UK: Cambridge University Press

Dae-Won Kim, Ki Young Lee, Doheon Lee and Kwang H Lee 2005 Evaluation of the performance of clustering algorithms in kernel-induced feature space. *Pattern Recognition* 38: 607–611

Dhillon, Yuqiang Guan and Brian Kulis 2005 *A Unified View of Kernel k-means, Spectral Clustering and Graph Cuts*. Technical Report, University of Texas, Department of Computer Sciences, Austin

Dhillon I, Guan Y and Kulis B 2004 Kernel k-means, spectral clustering and normalized cuts. In: *Proc. 10th ACM KDD Conference*, pages 1–6

Girolami Mark 2002 Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks* 13(3): 780–784

Heloise Hse and Richard Newton A 2004 Sketched symbol recognition using Zernike moments. In: *Proceedings of ICPR-2004*, volume 1, pages 367–370, Los Alamitos, CA, USA. IEEE Computer Society

Hitendra Sarma T and Viswanath P 2009 Speeding-up the k-means clustering method: A prototype based approach. In: *Proc. 3rd Int. Conf. on Pattern Recognition and Machine Intelligence(PReMI)LNCS 5909*, pages 56–61, Berlin Heidelberg: Springer-Verlag

Hitendra Sarma T, Viswanath P and Eswara Reddy B 2011 A fast and approximate kernel k-means clustering method for large datasets. In: *Proceedings of Intl. Conf. on Recent Advances in Intelligent Computational Systems(RAICS)-2011*, pages 545–550. IEEE

Hitendra Sarma T, Viswanath P and Eswara Reddy B 2012 A hybrid approach to speed-up the k-means clustering method. *International Journal of Machine Learning and Cybernetics(IJMLC)*, pages 1–11

Huaxiang Zhang and Jing Lu 2009 Semi-supervised fuzzy clustering: A kernel-based approach. *Knowledge-Based Systems* 22: 477–481

Hubert L and Arabie P 1985 Comparing partitions. *Journal of Classification* 2: 193–218

Jain A K, Murty M N and Flynn P J 1999 Data clustering: A review. *ACM Computing Surveys* 31(3): 264–323

Jan Flusser, Tomas Suk and Barbara Zitova 2009 *Moments and Moment Invariants in Pattern Recognition*. UK: John Wiley & Sons

Kim J, Shim K H and Choi S 2007 Soft geodesic kernel k-means. In: *Proc. 32nd IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 429–432

Klaus Robert Müller, Sebastian Mika, Gunnar Rätsch, Koji Tsuda and Bernhard Schölkopf 2001 An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks* 12(2): 181–201

Lon Bottou and Yoshua Bengio 1995 Convergence properties of the k-means algorithms. In: *Advances in Neural Information Processing Systems*, volume 7, pages 585–592. MIT Press

Maurizio Filippone, Francesco Camastra, Francesco Masulli and Stefano Rovetta 2008 A survey of kernel and spectral methods for clustering. *Pattern Recognition* (41): 176–190

Murphy P M 1994 *UCI Repository of Machine Learning Databases* [http://www.ics.uci.edu/mlearn/MLRepository.html]. Department of Information and Computer Science, University of California, Irvine, CA

Radha Chitta, Rong Jin, Timothy C Havens and Anil K Jain 2011 Approximate kernel k-means: Solution to large scale kernel clustering. In: *Proceedings of 17th ACM SIGKDD International Conference on Knowledge Discovery in Databases (KDD11)*, New York. ACM. doi:10.1145/2020408.2020558

Rand W M 1971 Objective criteria for the evaluation of clustering methods. *Journal of American Statistical Association* 66: 846–850

Ravindra Babu T and Narasimha Murty M 2001 Comparison of genetic algorithms-based prototype selection schemes. *Pattern Recognition* 34: 523–525

Richard O Duda, Peter E Hart and David G Stork 2000 *Pattern Classification*. UK: A Wiley-Interscience Publication, John Wiley & Sons, 2 edition

Scholkopf B, Smola A and Muller K R 1998 Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* 10: 1229–1319

Spath H 1980 *Cluster Analysis Algorithms for Data Reduction and Classification*. Chichester, UK: Ellis Horwood

Tzortzis Grigorios and Likas Aristidis 2008 The global kernel k-means clustering algorithm. *International Joint Conference on Neural Networks(IJCNN-08)*, pages 1978–1985

Wu Z D and Xie W X 2003 Fuzzy c-means clustering algorithm based on kernel method. In: *Proc. 5th Int. Conf. on Computational Intelligence and Multimedia Applications*, pages 1–6

Zhang R and Rudnicky A 2002 A large scale clustering scheme for kernel k-means. *ICPR-02*, pages 289–292