A Fast Approximate Kernel k-means Clustering Method For Large Data sets

T. Hitendra Sarma and P. Viswanath
Department of Computer Science and Engineering
Rajeev Gandhi Memorial College of Eng. and Technology
Nandyal-518501, A.P., India.
Email: {hitendrasarma, viswanath.p}@ieee.org

B. Eswara Reddy
Department of Computer Science and Engineering
JNTU College of Engineering
Anantapur-515002, A.P., India
Email:eswarcsejntu@gmail.com

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. The paper proposes a two stage hybrid approach to speed-up the kernel k-means clustering method. In the first stage, the data set is divided in to a number of group-lets by employing a fast clustering method called leaders clustering method. Each group-let is represented by a prototype called its leader. The set of leaders, which depends on a threshold parameter, can be derived in $O(n)$ time. The paper presents a modification to the leaders clustering method where group-lets are found in the kernel space (not in the input space), but are represented by leaders in the input space. 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 proposed method has time complexity of $O(n + p^2)$, where $p$ is the leaders set size. Its space complexity is also $O(n + p^2)$. The proposed method can be easily implemented. Experimental results shows that, with a small loss of quality, the proposed method can significantly reduce the time taken than the conventional kernel k-means clustering method.

I. 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. It has been considered as an important tool in various applications like pattern recognition, image processing, data mining, remote sensing, statistics, etc., [1]. 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, sum-of-squares based partitioning methods, like k-means clustering method, are effective. On the other hand, kernel based clustering methods, like kernel k-means clustering method, are proved to be effective to identify clusters which are non-isotropic and linearly inseparable in the input space [2] [3].

Girolami [3] first proposed the kernel k-means clustering method. It is an iterative method. It first maps the data points from the input space to higher dimensional feature space through a non linear 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 can be computed using a kernel function without knowing the explicit form of the transformation [4]. This is, because, the dot product between 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 $k : D \times D \to R$ is called the kernel function. This is often known as the kernel trick and is valid for transformations that satisfies Mercer’s conditions [5]. Some standard kernel functions are given below.

- Polynomial kernel, $k(x_i, x_j) = (x_i \cdot x_j + 1)^d$,
- Radial (RBF) kernel, $k(x_i, x_j) = \exp(-r||x_i - x_j||^2)$,
- Neural kernel, $k(x_i, x_j) = \tanh(ax_i \cdot x_j + b)$,

where $a$, $b$ and $d$ are positive constants.

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 $K = [k_{ij}]_{n \times n}$ 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. Other drawbacks being, (i) the number of clusters, $k$, should be given as input to the method, and (ii) the result is sensitive to the initial seed points, and the solution found may not be the optimal one because of the local minima problem.

Several improvements are proposed to cater these drawbacks. In order to overcome the local minima problem, Likas et al. proposed The global kernel k-means method [6], which produces a final partition which is independent of the initial seed points. Soft geodesic kernel k-means [7] method 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 labeled data. Kernel based approaches, such as, kernel based c-means method [8] ¹, kernel-based fuzzy c-means method, semi-supervised kernel fuzzy c-means method [9], etc., have been successfully used to deal with classification and clustering

¹Some authors call the k-means clustering method as the c-means clustering method.
problems. Spectral clustering methods are also used to identify non-linearly separable clusters in the input space. Some formal arguments show that, both kernel and spectral methods have some similarities [10][11]. Some other related methods are reviewed in [12].

Regarding the kernel k-means method for large data sets, recently a new scheme is proposed to overcome the problems of time and space complexities. In this scheme the kernel matrix is divided in to finite number of blocks such that size of each block is small enough to fit in the main memory and these blocks are processed in each iteration [13].

This paper presents a fast approximate kernel k-means clustering method suitable for large data sets. Our method runs in two stages. In the first stage, the data set is partitioned into blocks. In the second stage, kernel k-means clustering is applied with the set of leaders to derive a partition of the data set. The proposed hybrid approach runs faster than the conventional kernel k-means method, but, since the method is an approximate one, the quality of the final clustering result is slightly degraded. Also the space required to store the kernel matrix is significantly low in the proposed scheme. The proposed method is called prototype based hybrid kernel k-means clustering method.

The paper is organized as follows. Section II briefly reviews the kernel k-means clustering method while Section III outlines the leaders clustering method. The proposed method, prototype based hybrid kernel k-means clustering method, is described in Section IV. Experimental results are given in Section V and Section VI gives some of the conclusions and future work.

II. KERNEL K-MEANS CLUSTERING METHOD

Let \( D = \{x_1, x_2, \ldots, x_n\} \) be the data set of size \( n \), \( k \) be the number of clusters required, \( \mu^{(0)} \) be the initial seed points and \( \pi^{(0)} \) be the initial partition of the data set. Kernel k-means clustering method is an iterative method, takes \( k, \mu^{(0)} \) and an initial partition \( \pi^{(0)} \) as input and produces the final partition of the entire data set, \( \Pi_D \), as the output.

The objective function is to minimize the criterion function

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

where \( m_j \) is the mean of cluster \( C_j \). That is

\[
m_j = \frac{\sum_{\phi(x_i) \in C_j} \phi(x_i)}{|C_j|}
\]

in the induced space.

Distance between two data points \( \phi(x_i) \) and \( \phi(x_j) \) in the induced space, that is

\[
||\phi(x_i) - \phi(x_j)||^2 = \phi^2(x_i) - 2\phi(x_i) \cdot \phi(x_j) + \phi^2(x_j) = k(x_i, x_i) - 2k(x_i, x_j) + k(x_j, x_j).
\]  

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

\[
||\phi(x_i) - m_j||^2 = ||\phi(x_i) - \sum_{\phi(x_i) \in C_j} \frac{\phi(x_i)}{|C_j|}||^2 = \phi(x_i) \cdot \phi(x_i) - F(x_i, C_j) + G(C_j),
\]

where

\[
F(x_i, C_j) = -\frac{2}{|C_j|} \sum_{\phi(x_i) \in C_j} \phi(x_i) \cdot \phi(x_i)
\]

and

\[
G(C_j) = \frac{1}{|C_j|^2} \sum_{\phi(x_i) \in C_j} \sum_{\phi(x_s) \in C_j} \phi(x_i) \cdot \phi(x_s)
\]

The iterative method is outlined in the Algorithm 1.

Algorithm 1 kernel-k-means-clustering-method (\( D, k, \mu^{(0)}, \pi^{(0)} \))

<table>
<thead>
<tr>
<th>Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
</tr>
<tr>
<td>2.</td>
</tr>
<tr>
<td>3.</td>
</tr>
<tr>
<td>4.</td>
</tr>
<tr>
<td>5.</td>
</tr>
</tbody>
</table>

Output:
The final Partition \( \Pi_D = \{C_1, C_2, \ldots, C_k\} \).

III. KERNEL BASED LEADERS CLUSTERING METHOD

The paper presents a modification to the leaders clustering method, called, kernel based leaders clustering method. Using this method, the data set \( D \) is divided in to a number of group-lets in the induced space (not in the input space), but are represented by leaders in the input space. The number of group-lets depends on the threshold \( t \). Each group-let is represented by a prototype called its leader. For a given threshold distance \( t \), the kernel based leaders method maintains a set of leaders \( \mathcal{L} \). \( \mathcal{L} \) is initially empty and is incrementally built. For each data point \( x \) in the data set \( D \), if there is a leader \( l \in \mathcal{L} \), such that distance between \( \phi(x) \) and \( \phi(l) \) is less than or equal to \( t \), then \( x \) is assigned to the group-let represented by \( l \). Otherwise \( x \) itself becomes a new leader and is added to \( \mathcal{L} \). The algorithm outputs the set of leaders \( \mathcal{L} \). The leaders method is given in Algorithm 2.

IV. PROTOTYPE BASED HYBRID KERNEL K-MEANS CLUSTERING METHOD

This section describes the proposed hybrid kernel k-means clustering method. The method runs in two stages. First, a set of leaders \( \mathcal{L} \) is found using the modified leaders clustering method given in section III, later, 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. Let the size of the leaders set be \( p \). In our method, the running time taken for each iteration is significantly small, as it is working with only a few prototypes but not with the entire data set. The time complexity of conventional kernel k-means method is \( O(n^2) \). The time taken to generate the leaders in the induced space is \( O(n) \). The overall time complexity of the proposed hybrid method is \( O(n + p^2) \). Often, \( p \ll n \), hence the proposed method is faster. The space complexity of the method is also \( O(n + p^2) \). The proposed method is given in Algorithm 3.

The time and space complexities of our method depends on the number of leaders which in turn depends on the threshold \( t \). Increasing the \( t \) value can decrease the number of leaders \( p \), but, the deviation in the quality of the clustering result also might be increased. Experimental studies are done to show how \( t \) affects speed and quality.

Algorithm 2 kernel-based-leaders-clustering-method( \( \mathcal{D}, t \) )

\[
\mathcal{L} = \emptyset; \\
\text{for each } x \in \mathcal{D} \text{ do} \\
\quad \text{Find a } l \in \mathcal{L} \text{ such that } ||\phi(l) - \phi(x)|| \leq t \\
\quad \text{if there is no such } l \text{ or when } \mathcal{L} = \emptyset \text{ then} \\
\quad \quad \mathcal{L} = \mathcal{L} \cup \{x\}; \\
\text{end if} \\
\text{end for} \\
\text{Output } \mathcal{L};
\]

Algorithm 3 Prototype-based-kernel-k-means-method( \( \mathcal{D}, t \) )

1. Generate set of leaders \( \mathcal{L} \) by using the leaders clustering method given in Algorithm 2.
2. Compute the initial partition \( \pi^{(0)} \) of the leaders set \( \mathcal{L} \) using the given initial seed points \( \mu^{(0)} \).
3. Apply kernel-k-means-clustering-method (\( \mathcal{L}, k, \mu^{(0)}, \pi^{(0)} \)). Let \( \Pi_{\mathcal{L}} \) be the output.
4. Replace each leader \( l \), by its group-let in \( \Pi_{\mathcal{L}} \) to get the partition of the data set \( \Pi_\mathcal{D} \).
5. Output \( \Pi_\mathcal{D} \).

V. EXPERIMENTAL STUDY

This section describes the experiments performed with some bench mark data sets as well as with some synthetic data sets. The data sets employed are: Rings, Banana, Desert, Concentric Circles, Handwritten Symbols and Pendigits data sets. Pendigits data set is available at the UCI machine learning repository [15] and other data sets are artificially generated. The synthetic data sets used in our experiments includes various types of clusters. See Figure 1. All the data-sets have only numeric valued features. Handwritten digits data set is generated by collecting 16 different handwritten symbols from 10 different people. Each symbol is written on a 512 \( \times \) 512 canvas (writing pad). Each symbol is normalized to scale and translation, then Zernike moments of order 8 are extracted (since, it is found that Zernike moments of order 8 is giving better performance as for as classification is concerned). The properties of the data-sets are given in Table I.

Fig. 1. Two-dimensional artificial data sets used in the experiments, (a) Banana data set, (b) Rings data set, (c) Desert data set, and (d) Concentric Circles data set.

<table>
<thead>
<tr>
<th>Data-set</th>
<th>Number of data points</th>
<th>Number of Clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>BANANA</td>
<td>200</td>
<td>2</td>
</tr>
<tr>
<td>Rings</td>
<td>250</td>
<td>2</td>
</tr>
<tr>
<td>Concentric</td>
<td>230</td>
<td>2</td>
</tr>
<tr>
<td>Circles</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Desert</td>
<td>350</td>
<td>2</td>
</tr>
<tr>
<td>Handwritten</td>
<td>1600</td>
<td>16</td>
</tr>
<tr>
<td>Symbols</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PENDIGITS</td>
<td>10992</td>
<td>10</td>
</tr>
</tbody>
</table>

TABLE I
DETAILS OF DATA-SETS USED
Fig. 2. Clustering Accuracy(CA) and Running Time(RT) for Banana data set with respect to the threshold $t$. (a) Clustering Accuracy with respect to $t$ (b) Running Time with respect to $t$.

Fig. 3. Clustering Accuracy(CA) and Running Time(RT) for Rings data set with respect to the threshold $t$. (a) Clustering Accuracy with respect to $t$ (b) Running Time with respect to $t$.

<table>
<thead>
<tr>
<th>Data-set</th>
<th>CA using kernel k-means (In %)</th>
<th>Running Time (In seconds)</th>
<th>CA using our new method (In %)</th>
<th>Running Time (In seconds)</th>
<th>Percentage of reduction in Clustering Accuracy</th>
<th>Percentage of reduction in Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>BANANA</td>
<td>90.56</td>
<td>0.022</td>
<td>87.22</td>
<td>0.006</td>
<td>3.7</td>
<td>72.72</td>
</tr>
<tr>
<td>Rings</td>
<td>100</td>
<td>0.054</td>
<td>89.25</td>
<td>0.003</td>
<td>10.8</td>
<td>94.54</td>
</tr>
<tr>
<td>Concentric Circles</td>
<td>74.4</td>
<td>0.029</td>
<td>73.6</td>
<td>0.003</td>
<td>1.08</td>
<td>89.65</td>
</tr>
<tr>
<td>Desert</td>
<td>74.88</td>
<td>0.096</td>
<td>72.39</td>
<td>0.042</td>
<td>3.33</td>
<td>56.25</td>
</tr>
<tr>
<td>Handwritten Symbols</td>
<td>85.2</td>
<td>36.21</td>
<td>84.6</td>
<td>7.2</td>
<td>0.7</td>
<td>80.11</td>
</tr>
<tr>
<td>PENDIGITS</td>
<td>92.78</td>
<td>418.078</td>
<td>89.56</td>
<td>20.123</td>
<td>3.47</td>
<td>95.18</td>
</tr>
</tbody>
</table>
average percentage of reduction in running time is 81.41. The clustering accuracy and the overall running time depends on
the value of $t$. Figures 2, 3, 4, 5, 6, 7 shows the clustering accuracy and the total running time with respect to $t$ for all the data sets specified in table I. Each figure has two plots, where, the first one shows the relationship between CA and $t$, whereas, the second one shows the relationship between RT and $t$. From the presented results, it is observed that the proposed hybrid kernel k-means method runs in faster pace with a small loss of quality and hence it is applicable for large data sets.

VI. CONCLUSIONS AND FUTURE WORK

The paper presented a new hybrid approach to speed up the kernel k-means clustering method. In this hybrid approach, first, it finds a set of prototypes in the induced space. The prototypes are generated using kernel based leaders clustering method which takes linear time and the result of this method is used as input for kernel k-means method which produces a partition of prototypes. Finally, each prototype is replaced by its followers to get the partition of the entire data set. Our method runs in faster pace, however, there may be small loss in the quality of the clustering. The percentage of reduction in the accuracy can be reduced by selecting an appropriate value for the threshold $t$. This paper experimentally studied the effect of CA and RT with respect to $t$. Based on our experimental study, for $t$ approximately equal to 5% of average random pair wise distance, our new method gives high percentage of reduction in running time, with a small loss in clustering quality. The proposed method can be easily implemented. Future work is to give a theoretical way of fixing the appropriate $t$ value.

ACKNOWLEDGMENTS

The work reported in this paper is supported by a sponsored project from “All India Council for Technical Education - New Delhi” under RPS scheme with reference: “F.No: 8023/BOR/RID/RPS-51/2009-10”.

REFERENCES


