Pattern Recognition 44 (2011) 2862–2870

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A distance based clustering method for arbitrary shaped clusters in large datasets

Bidyut Kr. Patra [a,*], Sukumar Nandi [a], P. Viswanath [b]

[a] Department of Computer Science and Engineering, Indian Institute of Technology – Guwahati, Guwahati 781039, India
[b] Department of Computer Science and Engineering, Rajeev Gandhi Memorial College of Engineering & Technology, Nandyal 518501, A.P., India

ARTICLE INFO

Article history:
Received 12 September 2009
Received in revised form
16 April 2011
Accepted 29 April 2011
Available online 13 May 2011

Keywords:
Distance based clustering
Arbitrary shaped clusters
Leaders
Single-link
Hybrid clustering method
Large datasets

ABSTRACT

Clustering has been widely used in different fields of science, technology, social science, *etc*. Naturally, clusters are in arbitrary (non-convex) shapes in a dataset. One important class of clustering is distance based method. However, distance based clustering methods usually find clusters of convex shapes. Classical single-link is a distance based clustering method, which can find arbitrary shaped clusters. It scans dataset multiple times and has time requirement of $O(n^2)$, where n is the size of the dataset. This is potentially a severe problem for a large dataset. In this paper, we propose a distance based clustering method, *l*-SL to find arbitrary shaped clusters in a large dataset. In this method, first leaders clustering method is applied to a dataset to derive a set of leaders; subsequently single-link method (with distance stopping criteria) is applied to the leaders set to obtain final clustering. The *l*-SL method produces a flat clustering. It is considerably faster than the single-link method applied to dataset directly. Clustering result of the *l*-SL may deviate nominally from final clustering of the single-link method (distance stopping criteria) applied to dataset directly. To compensate deviation of the *l*-SL, an improvement method is also proposed. Experiments are conducted with standard real world and synthetic datasets. Experimental results show the effectiveness of the proposed clustering methods for large datasets.

1. Introduction

Clustering problem appears in many different fields like data mining, pattern recognition, statistical data analysis, bio-informatics, *etc*. Clustering problem can be defined as follows. Let $\mathcal{D}=\{x_1,x_2,x_3,\ldots,x_n\}$ be a set of n patterns, where each x_i is a N-dimensional vector in a given feature space. Clustering activity is to find groups of patterns, called clusters in $\mathcal{D}$, in such a way that patterns in a cluster are more similar to each other than patterns in distinct clusters. There exist many clustering methods in the literature [1–4].

Clustering methods are mainly divided into two categories based on the way they produce results *viz*., partitional clustering and hierarchical clustering methods.

Partitional clustering methods create a single clustering (flat clustering) of a dataset. Partitional methods can be further categorized into two classes based on the criteria used *viz*., *distance based* and *density based*. Distance based methods optimize a global criteria based on the distance between patterns. Some of the popular distance based clustering methods are *k*-means [5], CLARA [6], and CLARANS [7]. Density based partitional clustering methods optimize local criteria based on density distribution of patterns. DBSCAN [8] and DenClue [9] are of this type of clustering methods.

Hierarchical clustering methods create a sequence of nested clusterings of a dataset. These methods can also be categorized into two classes *viz*., density based and distance based. Clustering methods like single-link [10], complete-link [11], and average-link [12] are distance based hierarchical clustering methods. Clustering methods like OPTICS [13] and Chameleon [14] are density based hierarchical clustering methods.

Single-link clustering method can find arbitrary shaped clusters in many applications such as image segmentation, spatial data mining, geological mapping. It builds a dendogram where each level represents a clustering of the dataset. A suitable clustering is chosen from the dendogram based on requirements. Selection of a clustering from the dendogram can be done by specifying various stopping conditions as follow [15]:

(a) *Distance-h stopping condition*: Distance between any pair of clusters in the clustering should be greater than h.
(b) *k-cluster stopping condition*: Number of clusters in the clustering should be k.

* Corresponding author.
E-mail addresses: bidyut@iitg.ernet.in (B.K. Patra), sukumar@iitg.ernet.in (S. Nandi), viswanath.pulabaigari@gmail.com (P. Viswanath).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.04.027

(c) *Scale-α stopping condition*: Let ρ^* be the maximum pairwise distance between any two patterns in the dataset. Then, the stopping condition is: distance between any pair of clusters should be greater than $\alpha\rho^*$, where $0<\alpha<1$.

In this paper, we consider single-link clustering method with distance-*h* stopping condition and it is referred to as *SL* method in the remaining part of the paper. In the study [16], Krause exploited SL method in computational biology. However, SL method has the following drawbacks: (i) time complexity of $O(n^2)$ and (ii) scanning the dataset many times. Therefore, this method is not suitable for large datasets.

In this paper, we propose a distance based clustering method which is suitable for clustering large datasets by combining leaders clustering method (partitional clustering) and SL method. We call this hybrid method as *leader-single-link* (*l*-SL) method, (*l* stands for leaders). The *l*-SL method is significantly faster than the SL method. However, the clustering results produced by *l*-SL method may deviate from the final clustering produced by the SL method. From various experimental studies, it is found that deviations are nominal. To overcome this deviation (if any), a correction scheme is also proposed and we call it as *al*-SL method. Execution time of the *al*-SL method is marginally higher than the *l*-SL method, but faster than the SL method. In this paper, we also prove that clustering results produced by the *al*-SL method and the SL method are identical. Experimental results with standard real world and synthetic datasets show the effectiveness of the proposed clustering methods in large datasets.

The rest of the paper is organized as follows. Section 2 describes a summary of related works. Section 3 describes a brief background of the proposed clustering methods. Section 4 describes proposed distance based clustering method (*l*-SL) and a relationship between SL and *l*-SL methods. Section 5 describes proposed improvement of the *l*-SL method. Experimental results and conclusions are discussed in Section 6 and Section 7, respectively.

2. Related work

In this section, recent works related to the proposed schemes are discussed. Dash et al. [17] proposed a fast hierarchical clustering method based on partially overlapping partitions (POP). It has two phases. In first phase, dataset is partitioned into a number of overlapping cells. Closest pair-distance is calculated for each cell and if overall closest pair-distance is less than a threshold δ, then the pair is merged. This process continues until overall closest pair distance is more than δ. In second phase, traditional hierarchical agglomerative clustering (HAC) method is applied over the remaining clusters. However, HAC uses centroid based metric to calculate distance between clusters.

Nanni [18] exploits triangle inequality property of metric space to speed-up hierarchical clustering method (single-link and complete-link). Considering the distribution of data, the scheme reported gain in the reduction of distance computations for a variant of single-link method (*k-cluster stopping condition*). However, these methods keep entire dataset in main memory for processing. So these methods are not suitable for large datasets.

Vijaya et al. [19] proposed a hybrid clustering technique to speed up protein sequence classification. In this method, initially leaders clustering is applied to dataset and traditional HAC clustering methods are applied to the leaders at subsequent steps to obtain *k* clusters. A median of each of *k* clusters is selected as cluster representative. To classify a test pattern, a classifier selects the closest cluster representative and subsequently closest sub-cluster in that cluster. In this method, the hybrid clustering technique is used to reduce computational cost of the classifier only. However, no theoretical and experimental analysis of clustering results are reported in the paper. Relation between leaders threshold (τ) and the number of cluster representatives (k) are also not reported.

Recently, Koga et al. [20] proposed a fast approximation algorithm called *LSH-link* which is a faster way of implementing single-link method. Unlike traditional single-link method, it quickly finds nearer clusters in linear time using a probabilistic approach (*Locality-Sensitive Hashing* [21]). They claimed that their method ran in linear time under certain assumptions. However, it keeps entire dataset in main memory. This prohibits its application to large datasets. Clustering results are highly influenced by parameters (number of hash functions, number of entries in hash tables, initial threshold distance, ratio of two successive layer-distances of dendogram). For high dimensional dataset, size of hash table could be very large and unmanageable.

Few other clustering methods had been proposed to combine partitional clustering method with hierarchical clustering method to get the advantages of both methods [22–25]. Clustering method proposed in [22] is a multilevel method, in which *k*-means is combined with single-link clustering method. Clustering results are empirically analyzed. Clustering methods proposed in [23–25] divide dataset into a number of sub-clusters applying *k*-means method; subsequently an agglomerative method is applied to sub-clusters for final clustering. These methods differ in measuring similarity (separation) between a pair of sub-clusters. The hybrid method in [23] can identify high density clusters in one dimensional dataset. However, there is no generalized analysis for multi dimensional dataset. All these methods use density information in measuring similarity between a pair of clusters. Therefore, these methods are not discussed further in this paper. It is noted that few works reported theoretical analysis only from computational point of view [22,23]. However, clustering results obtained using hybrid methods are not theoretically analyzed with the clustering output of the single-link method.

3. Background of the proposed method

Two widely used clustering methods *viz.*, leaders clustering and single-link clustering method are discussed in this section. These two clustering methods have their own advantages and disadvantages. The proposed clustering methods exploit these two clustering methods.

3.1. Leaders clustering method

Leaders clustering method [26,27] is a single data-scan distance based partitional clustering method. Recently, leaders clustering method has been used in pre-clustering phase in data mining applications [28,29]. For a given threshold distance τ, it produces a set of leaders $\mathcal{L}$ incrementally. For each pattern x, if there is a leader $l\in\mathcal{L}$ such that $\|x-l\|\leq\tau$, then x is assigned to a cluster represented by l. In this case, we call x as a *follower* of the leader l. If there is no such leader, then x becomes a new leader. Each leader can be seen as a representative of a cluster of the patterns which are grouped with it. The time complexity of leaders clustering method is $O(mn)$, where $m=|\mathcal{L}|$. The space complexity is $O(m)$, if only leaders are stored. Otherwise, it is $O(n)$. However, it can only find convex shaped clusters. It is an order dependent clustering method.

3.2. Single-link clustering method

Single-link [10,30] is a distance based agglomerative hierarchical clustering method. In single-link, distance between any two clusters C_1 and C_2 is the minimum of distances between all pairs in $C_1 \times C_2$[1]. That is,

$$Distance(C_1,C_2) = \min\{\|x_i - x_j\| \mid x_i \in C_1, x_j \in C_2\}$$

It produces a hierarchy of clusterings. In this paper, we consider minimum inter-cluster distance (h) as the selection criteria to find the final clustering from the hierarchy. The single-link method with inter-cluster distance (h) is noted as the SL method and depicted in Algorithm 1. The SL method is not scalable with the size of dataset. The time and space requirements of SL method are $O(n^2)$ and it scans the dataset many times.

Algorithm 1. SL ($\mathcal{D}$, h).

{$\mathcal{D}$ is a given dataset, h is cut-off distance}
Place each pattern $x \in \mathcal{D}$ in a separate cluster. Let $\pi_1 = \{C_1, C_2, \ldots, C_n\}$
Compute the inter-cluster distance matrix and set $i=1$.
while {A pair of clusters $C_x, C_y \in \pi_i$ such that *Distance* $(C_x, C_y) \le h$}
  Select two closest clusters C_l and C_m.
  Form a new cluster $C = C_l \cup C_m$.
  Next clustering is $\pi_{i+1} = \pi_i \cup \{C\} \backslash \{C_l, C_m\}$; $i = i+1$
  Update distances from C to all other clusters in the current clustering π_i.
end while
Output final clustering π_i.

4. Proposed clustering method

The proposed l-SL clustering method is a hybrid scheme with a combination of above two techniques (i.e. leaders and SL method). The l-SL method needs only a single parameter h (i.e. distance between a pair of clusters is more than h). Like other clustering methods (k-means, DBSCAN), we assume that the value of the parameter (h) is known before hand. In this scheme, the set of leaders produced by the leaders clustering method is used as a representative set of data. Subsequently, the SL method is applied to the representative set to obtain final clustering.

4.1. Selection of leaders clustering method and its threshold

There exists a category of clustering method called sequential algorithm [31]. These methods are fast and create a single clustering of a dataset. Leaders clustering method [26,27] is one of this type. Basic Sequential Algorithmic Scheme (BSAS), modified BSAS (MBSAS) [31], Max-min algorithm [32], Two-Threshold Sequential Algorithmic Scheme (TTSAS) [31] are different variations of the leaders clustering method. However, all these methods either scan dataset more than once (MBSAS, TTSAS) or need more than one parameters (BSAS, TTSAS).

The k-means [5] method runs in linear time with the size of dataset. It can be used to select prototypes of a dataset. However, k-means is applicable to numeric datasets only and scans dataset more than once before convergence.

The leaders clustering method can be applied to any datasets where notion of distance (similarity) is defined. It needs one parameter (τ) and single database scan.

[1] This notation of distance between two sets of points will be used subsequently.

Distance criteria τ for the leader clustering method in l-SL method plays a major role to produce final clustering. It is obvious that τ should be less than that of h. It is observed from experimental results that $\tau \le h/2$ is required to get clustering results at par with results of SL method applied directly. Minimum execution time of l-SL method is found for $\tau = h/2$. Results obtained using $\tau = h/2$ is explained analytically in subsequent subsection.

4.2. The l-SL method: a variant of single-link clustering method

The l-SL method works as follows. At first, a set of leaders ($\mathcal{L}$) is obtained applying the leaders clustering method to a dataset using $\tau = h/2$. The leaders set is further clustered using SL method with cut-off distance h which results in clustering of leaders. Finally, each leader is replaced by its followers to produce final clustering. The l-SL method is depicted in Algorithm 2.

Algorithm 2. l-SL ($\mathcal{D}$, h).

{$\mathcal{D}$ is a given dataset, h is cut-off distance}
Apply leaders method with $\tau = \frac{h}{2}$ to $\mathcal{D}$.
{Let $\mathcal{L}$ be the output }
Apply SL ($\mathcal{L}$,h) as given in Algorithm 1.
{Outputs a clustering $\pi_{\mathcal{L}}$ of the leaders.}
Each leader in $\pi_{\mathcal{L}}$ is replaced by its followers.
{This gives a clustering of $\mathcal{D}$ (say π_l).}
Output π_l.

Time and space complexity of the proposed method are analyzed as follows:

1. Step of obtaining the set of leaders ($\mathcal{L}$) takes time of $O(mn)$, where $m = |\mathcal{L}|$. Space complexity is $O(m)$. It scans dataset once.
2. The SL ($\mathcal{L}$, h) has same time and space complexity of $O(m^2)$.

Overall running time of l-SL is $O(mn+m^2)$. Since, m is always less than n, the complexity becomes $O(mn)$. Experimentally, we show that l-SL is considerably faster than that of SL method. It is because, SL works with the entire dataset, whereas l-SL works with the leaders set (a subset of the dataset). Space complexity of l-SL method is $O(m^2)$.

4.3. Relationship between SL and l-SL

Relationship between SL method and l-SL method is formally established in this section. For $x,y \in \mathcal{D}$ and a clustering π, we denote $x \sim_{\pi} y$ if x and y are in a same cluster of π and $x \nsim_{\pi} y$, otherwise.

Let π and π_l be final clustering produced by SL (distance stopping criteria) and l-SL for same value of h, respectively. Then, $\pi = \pi_l$, if the following two conditions hold.

1. For $x,y \in \mathcal{D}$, if $x \sim_{\pi} y$, then $x \sim_{\pi_l} y$.
2. For $x,y \in \mathcal{D}$, if $x \nsim_{\pi} y$, then $x \nsim_{\pi_l} y$

Following definitions, lemmas and theorem are demonstrated for applicability of above two conditions for π and π_l.

Definition 4.1 (*Refinement*). [33] A refinement of a partition $\pi_1 = \{B_1^{(1)}, B_2^{(1)}, \ldots, B_p^{(1)}\}$ of $\mathcal{D}$ is a partition $\pi_2 = \{B_1^{(2)}, B_2^{(2)}, \ldots, B_q^{(2)}\}_{q \ge p}$ of $\mathcal{D}$ such that for each $B_j^{(2)} \in \pi_2$ there is a $B_i^{(1)} \in \pi_1$ and $B_j^{(2)} \subseteq B_i^{(1)}$.

Lemma 4.1. *If two arbitrary patterns x and y are in a dataset and SL method satisfies $x \nsim_{\pi} y$ then l-SL satisfies $x \nsim_{\pi_l} y$.*

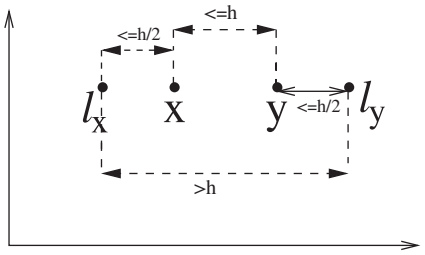

Fig. 1. A probable representation for Lemma 4.2.

Proof. This is proved by contradiction. Any two patterns x and y are not in a cluster according to SL method (i.e. $x \nsim_{\pi} y$) means that there is no sequence of patterns $x,a_1,\ldots,a_p,y$ such that distance between any two successive patterns is less than or equal to h. If this does not hold then x and y must be in a cluster.

Let l_x be a leader such that $x \in followers(l_x)$, similarly let l_y be a leader such that $y \in followers(l_y)$. Suppose there is a sequence of leaders $l_x,l_1,\ldots,l_q,l_y$ such that distance between any two successive leaders in the sequence is less than or equal to h. If so, l_x and l_y are grouped into a cluster according to the l-SL. Since, a leader is also a pattern in a dataset, there is a sequence of patterns $x,l_x,l_1,\ldots,l_q,l_y,y$ such that distance between any two successive patterns is less than or equal to h. This is true for l-SL method, since distance between a pattern and its leader is always less than or equal to $h/2$. So, x and y must be grouped in a cluster according to SL. This is the required contradiction. □

Lemma 4.2. *If two arbitrary patterns x and y are in a dataset and SL method satisfies $x \sim_{\pi} y$, then l-SL method may not satisfy $x \sim_{\pi_l} y$.*

Proof. The following scenario may exist in final clustering. Let x be a follower of a leader l_x and y be a follower of a leader l_y. If $\|x-y\| \le h, \|l_x-x\| \le h/2, \|l_y-y\| \le h/2$ and $\|l_x-l_y\| > h$, clearly, SL method groups these four patterns into a cluster. Since $\|l_x-l_y\| > h$, l_x,l_y and their followers are not grouped into a cluster by l-SL (for clarity, this is also shown in Fig. 1). Hence, $x \nsim_{\pi_l} y$ holds. □

Lemma 4.1 states that if x and y are not grouped into a cluster by SL method then l-SL does not group them into a cluster. Lemma 4.2 states that the number of clusters in the final clustering produced by l-SL method may be more than that of the SL method, i.e. few clusters produced by l-SL method might be grouped as one cluster in SL method. From the above two lemmas the following Theorem 4.1 can be derived.

Theorem 4.1. *The clustering output of l-SL method is a refinement of the clustering output of SL method.*

As per the Theorem 4.1, the cardinality of final clustering generated by l-SL method may be more compared to final clustering generated by SL method for a given dataset, if they consider the same value of h. Experiments with various standard and synthetic datasets show that the deviation (i.e. difference in cardinality) is nil or a small value. However, to get exact clustering as that of the SL method, further refinements (i.e. merging of the deviated clusters) of l-SL method is proposed in the following section.

5. Augmented *l*-SL (*al*-SL) clustering method

As discussed earlier, a few clusters in the final results of l-SL method may be required to be merged in order to obtain the exact final results as that of the SL method. Let the clustering result of the l-SL be a clustering $\pi_l = \{B_1,B_2,\ldots,B_k\}$. Let (B_i,B_j) be a pair of clusters in π_l. If there exists a pair of patterns $x \in B_i$ and $y \in B_j$ such that $\|x-y\| \le h$, then B_i and B_j must be merged together. In general, one needs to search for all pairs of clusters in π_l to see whether they can be merged. The following Lemma 5.1 shows that it is not required to search all pairs of clusters, except for a few pairs.

Definition 5.1 (*Cluster of leaders*). A cluster of leaders $B_i^l = \{l_1, l_2,\ldots,l_k\}$ is a group of k leaders, which are found by the l-SL method.

Lemma 5.1. *Let B_i^l and B_j^l be two clusters of leaders as found by the l-SL method which are expanded into clusters of patterns (i.e. each leader is replaced with its followers set) B_i and B_j. If Distance $(B_i^l,B_j^l) > 2h$, then Distance $(B_i,B_j) > h$.*

Proof. Let l_i be an arbitrary leader in B_i^l and l_j be an arbitrary leader in B_j^l such that $\|l_i-l_j\| > 2h$. Let x be an arbitrary pattern in $followers(l_i)$ and y be an arbitrary pattern in $followers(l_j)$.

As x is a follower of l_i, we get $\|x-l_i\| \le h/2$. Since $\|x-l_i\| \le h/2$ and $\|l_i-l_j\| > 2h$, we get $\|x-l_j\| > 1.5h$ (based on triangle inequality).

Similarly, $\|l_j-y\| \le h/2$ and $\|x-l_j\| > 1.5h$, we get $\|x-y\| > h$. Considering $\|x-y\| > h$ for all x and y, it can be concluded that *Distance* $(B_i,B_j) > h$. □

It is obvious (from the Lemma 5.1) that if *Distance* $(B_i^l,B_j^l) > 2h$, merging of the clusters B_i and B_j is not possible. On the other hand, as per l-SL method *Distance* $(B_i^l,B_j^l) \le h$ is not possible (i.e. both of the clusters are in same cluster). So, the clusters B_i^l and B_j^l may be merged, if $h < Distance(B_i^l,B_j^l) \le 2h$. Therefore, followers of a few leaders in B_i^l and B_j^l need to be searched for possible merging of clusters. It can be mentioned that merging of clusters pair (B_i,B_j) is equivalent to merging of the clusters (B_i^l,B_j^l). Considering all these points, we have augmented l-SL method by incorporating an option of merging of clusters. This method is termed as *al*-SL method.

Once clustering of leaders $(\pi_{\mathcal{L}})$ is available from l-SL method, *al*-SL method finds pairs of clusters (B_i^l,B_j^l) such that $Distance(B_i^l,B_j^l) \le 2h$. Let (l_i,l_j), $l_i \in B_i^l, l_j \in B_j^l$ be a pair of leaders such that $\|l_i-l_j\| = Distance(B_i^l,B_j^l)$. Next, we identify all leaders $l_x \in B_i^l$ and $l_y \in B_j^l$ which satisfy following conditions: (i) $\|l_i-l_y\| \le 2h$ and (ii) $\|l_j-l_x\| \le 2h$. These leaders l_x,l_y are the potential leaders for possible merging of the clusters (B_i^l,B_j^l).

All such leaders are grouped for further operations. Let L_{B_i} be a set of all l_x leaders and L_{B_j} be a set of all l_y leaders. It implies that followers of the leaders in sets L_{B_i} and L_{B_j} are potential decision maker for merging the cluster-pair B_i^l and B_j^l. The next step is to find all followers of these leaders, this requires the leaders clustering to be executed once more (ordering of patterns in dataset and $\tau = h/2$ to be same as the first time scanning). This time we store followers of all potential leaders for merging.

The last step of possible merging is to find the distance between pair of patterns (i.e. followers) of a pair of clusters (B_i^l,B_j^l) such that $Distance(B_i^l,B_j^l) \le 2h$. If the distance between two patterns (i.e. followers) of the pair of clusters is less than or equal to h, then the cluster-pair is merged. The *al*-SL method is given in Algorithm 3.

Algorithm 3. *al*-SL ($\mathcal{D}$, h).

{$\mathcal{D}$ is a given dataset, h is cut-off distance}
Apply leaders method with $\tau = h/2$ to $\mathcal{D}$.
{Outputs $\mathcal{L}$, set of leaders.}
Apply SL ($\mathcal{L}$,h) as given in Algorithm 1.
{Outputs $\pi_{\mathcal{L}}$, a clustering of $\mathcal{L}$.}
$\mathcal{S} = \emptyset$ {*The merging part begins*}
for each pair of clusters (B_i^l,B_j^l) in $\pi_{\mathcal{L}}$ with $Distance(B_i^l,B_j^l) \le 2h$
do
Identify a pair of leaders (l_i,l_j) such that
$\|l_i-l_j\| = Distance(B_i^l,B_j^l)$)

$L_{B_i}=\emptyset$; $L_{B_j}=\emptyset$
for each $l_x \in B_i^l$ **do**
  if $\|l_x-l_j\| \le 2h$ **then**
    $L_{B_i}=L_{B_i}\cup\{l_x\}$
  end if
end for
for each $l_y \in B_j^l$ **do**
  if $\|l_y-l_i\| \le 2h$
    $L_{B_j}=L_{B_j}\cup\{l_y\}$
  end if
end for
$\mathcal{S}=\mathcal{S}\cup L_{B_i}\cup L_{B_j}$. {$\mathcal{S}$, set of potential leaders responsible for possible merging}
end for
if $\mathcal{S}\neq\emptyset$ **then**
Apply leaders method with $\tau=h/2$ and store followers of leaders in $\mathcal{S}$. {$\mathcal{D}$ *is scanned for second time in same order*}
for each pair of clusters (B_i^l,B_j^l) in $\pi_{\mathcal{L}}$ such that $Distance(B_i^l,B_j^l)\le 2h$ **do**
  for each pair of potential leaders (l_a,l_b) such that $l_a\in L_{B_i}$ and $l_b\in L_{B_j}$ **do**
    Find two nearest followers (x,y); $x\in l_a$ and $y\in l_b$
    if $\|x-y\|\le h$ **then**
      Merge B_i^l and B_j^l into a single cluster; break;
    end if
  end for
end for
end if {*The merging part ends*}
Each leader in $\pi_{\mathcal{L}}$ is replaced by its followers. Let this clustering be π_{alSL}.
Output π_{alSL}.

5.1. Complexity analysis

The *al*-SL method has two parts *viz.*, *l*-SL part and the merging part. The complexity analysis of the *l*-SL is discussed in previous section. Here, we discuss complexity of the merging part as follows.

1. Finding potential leaders sets for possible merging of a pair of clusters (B_i^l,B_j^l) can be done in $O(m)$ time. For all pairs of clusters, it takes a time of $O(m)$. Space requirement for this step is $O(m^2)$.
2. Space requirement to store the followers of $\mathcal{S}$ is $O(\sum_{k=1}^{|\mathcal{S}|}|l_k|, l_k\in\mathcal{S})$ and the time requirement is $O(mn)$.
3. Time required for finding minimum distance between a pair of leaders is $O(f^2)$, where f is the average number of followers of a leader. If m_a is the average number of potential leaders responsible for merging a pair of clusters, then the time required to find minimum distances for all pair of clusters is $O(m_a^2f^2)$.

Time complexity of the merging part is $O(m)+O(mn)+O(m_a^2f^2)$, i.e. $O(mn+m_a^2f^2)$ as $m<n$. As $m_a<m, m_a^2f^2\approx mn$. So, time complexity of merging part is approximately $O(mn)$. Space complexity is $O(m^2)+O(|\mathcal{S}|*f)\approx O(m^2)$ (since, $|\mathcal{S}|$ is very less than m). Overall time complexity of *al*-SL method becomes $O(mn)$ and overall space complexity is $O(m^2)$.

As we discussed in Section 3.1, the results of leaders clustering depend on the scanning order of the dataset. The same is true for *l*-SL method. However, clustering results of *al*-SL method are independent of the scanning order. It is proved in the following subsection 5.2.

5.2. Relationship between SL and al-SL

In this section, we formally establish that clustering result produced by the *al*-SL method is same as that of the SL method and result of the *al*-SL is independent of the scanning order of the dataset.

Definition 5.2 (*Reachable*). A pattern $x\in\mathcal{D}$ is reachable from another pattern $y\in\mathcal{D}$, if there exists a sequence of patterns $x_1,x_2,\ldots,x_p$, where $x_1=y$, $x_p=x$, and $\|x_i-x_{i+1}\|_{i=1..p-1}\le h, x_i\in\mathcal{D}$.

This *reachable* relation $R\in\mathcal{D}\times\mathcal{D}$ is an equivalence relation.

Lemma 5.2. *The clustering results produced by the al-SL method is same as that of the SL method.*

Proof. Let $\pi_{SL}=\{C_1,C_2,\ldots,C_k\}$, where $C_i\subseteq\mathcal{D}$, be the clustering result produced by SL method. $C_i=\{x_i\in\mathcal{D}: x_i$is reachable from $x_j\in C_i\}$ is a cluster of $\mathcal{D}$. The π_{SL} is a partition of $\mathcal{D}$. Therefore, there exists a unique equivalent relation R_1 on $\mathcal{D}$ and each $C_i\in\pi_{SL}$ is an equivalence class of $\mathcal{D}$ by the relation $R_1, C_i=[x_i]_{R_1}=\{x_j\in\mathcal{D}\mid x_iR_1x_j\}$. Clearly, R_1 is a reachable relation of $\mathcal{D}$.

The *al*-SL method initially creates a partition $\pi_{\mathcal{L}}=\{B_1^l,B_2^l,\ldots,B_p^l\}$ of leaders set $\mathcal{L}$, where B_i^l is a cluster of leaders. $\pi_l=\{B_1,B_2,\ldots,B_p\}$ can be a partition of the dataset if each leader $l\in B_i^l\in\pi_{\mathcal{L}}$ is replaced by its followers (Note that each B_i^l converted into $B_i\in\pi_l$ with a mapping $f:\pi_{\mathcal{L}}\to\pi_l$ such that $f(B_i^l)=B_i$). However, $\pi_l\neq\pi_{SL}$ due to Theorem 4.1. There may exist a pair of patterns $x\in B_i, y\in B_j (i\neq j)$ such that $\|x-y\|\le h$ (Lemma 4.2). Again, according to Lemma 5.1, there cannot be any pair of patterns $x\in B_i, y\in B_j$ such that $\|x-y\|\le h$, if $Distance(B_i^l,B_j^l)>2h$. Therefore, *al*-SL exhaustively searches followers of all potential leaders of each pair (B_i^l,B_j^l) with $Distance(B_i^l,B_j^l)\le 2h$ (Algorithm 3). If there exists a pair of followers (patterns) x,y of any potential leaders pair of (B_i^l,B_j^l) such that $\|x-y\|\le h$, then *al*-SL method merges (B_i^l,B_j^l). Finally, it produces a clustering $\pi_{alSL}=\{B_1,B_2,\ldots,B_k\}_{k<=p}$. The π_{alSL} is also a partition of $\mathcal{D}$ and corresponds an equivalence relation R_2. Therefore, $B_i\in\pi_{alSL}$ is an equivalent class of $\mathcal{D}$ by R_2. More formally, $[x_i]_{R_2}=B_i=\{x_j\in\mathcal{D}: x_j$ is reachable from $x_i\in B_i\}$. Clearly, R_2 is also a reachable relation of $\mathcal{D}$. Therefore, $R_2=R_1$. It follows that $\pi_{SL}=\pi_{alSL}$. □

Theorem 5.1. *Clustering results produced by the al-SL method is independent of the scanning order of the dataset by leaders clustering method.*

Proof. For the sake of simplicity, we consider two different scanning orders of the dataset $\mathcal{D}$ by leaders clustering method. Let $\mathcal{L}=\{l_1,l_2,\ldots,l_{m1}\}$ and $\mathcal{L}'=\{l'_1,l'_2,\ldots,l'_{m2}\}$ be two sets of leaders obtained in two different scanning orders. Having applied the *al*-SL method to $\mathcal{L}$ and $\mathcal{L}'$ separately, we obtain two partitions $\pi_{alSL}=\{B_1,B_2,\ldots,B_k\}$ and $\pi'_{alSL}=\{B'_1,B'_2,\ldots,B'_{k'}\}$, respectively. We have to show that $\pi_{alSL}=\pi'_{alSL}$.

From Lemma 5.2, we can say that π_{alSL} and π'_{alSL} both correspond to same equivalence relation, which is a reachable relation on $\mathcal{D}$. Therefore, $\pi_{alSL}=\pi'_{alSL}$. □

5.3. Estimating the value of h

Our clustering methods *l*-SL and *al*-SL need a parameter h. Two approaches for finding the value of h are reported as follow.

- *First approach*: Many *domain experts* know approximate distances between natural clusters. For example, cell biologists might have an idea of average distance between the chromosomes. Medical

practitioner might have some knowledge about the average distances between bones in different parts of the body in x-ray image.

- *Second approach*: We select randomly $O(\sqrt{n})$ patterns from a dataset and single-link clustering method is applied to these selected patterns. From dendogram generated by the single-link method, one can find *life time* for each clustering. *Life time* (*LT*) [34] of a clustering π_i is defined as follows:

  $LT(\pi_i) = (d_i - d_{i+1})$,

  where d_i, d_{i+1} are the distances between two closest clusters in π_i and π_{i+1} clusterings, respectively.
  π_i, π_{i+1} are clusterings obtained from two consecutive layers of the dendogram. *Maximum life time* clustering is that for which life time is maximum.
  This *maximum life time* can be considered as the value of h.

6. Experimental evaluation

In this section, we evaluate the proposed methods (*l*-SL and *al*-SL) experimentally. From discussion in Section 2, it is clear that scheme proposed in [17] is not directly related to our schemes. The method proposed in [18] keeps dataset as well as distance-matrix for the entire data in primary memory. The scheme proposed in [20] needs many parameters to be adjusted to produce same clustering as that of the single-link method, which are difficult to determine. This method also requires whole dataset to be kept in primary memory. Therefore, these methods are not suitable for large datasets. The hybrid clustering method in [19] is used to reduce computational burden of classification. Clustering results are not analyzed experimentally. There is no relation between the leaders threshold (τ) and the number of clusters (k). Therefore, comparisons with the proposed schemes are not reported here.

So, clustering results of the *l*-SL, *al*-SL methods with that of the SL method are compared with help of Rand Index (*RI*) [35]. Rand Index (*RI*) is defined as follows.

Given a dataset $\mathcal{D}$ of n patterns and let π_1 and π_2 be two clusterings of $\mathcal{D}$. *RI* is a similarity measure between a pair of clusterings, i.e.

$$RI(\pi_1,\pi_2) = \frac{a+d}{a+b+c+d}$$

where a is the number of pairs of patterns belonging to a cluster in π_1 and to a same cluster in π_2, b is the number of pairs belonging to a same cluster in π_1 but to different clusters in π_2, c is the number of pairs belonging to different clusters in π_1 but to a same cluster in π_2, and d is the number of pairs of patterns belonging to different clusters in π_1 and to different clusters in π_2.

We compute Rand Index (*RI*) between *l*-SL and SL method, and between *al*-SL and SL method for various values of h. To prove effectiveness of the proposed methods, experiments are conducted with standard as well as large datasets. Two synthetic and five real world datasets (http://archive.ics.uci.edu/ml/, http://www.ncbi.nlm.nih.gov/geo/) are used after eliminating class labels. A brief description of the datasets is given in Table 1. Plot of a synthetic data (Spiral Data) is shown in Fig. 2.

6.1. Experiments with standard datasets

All methods (*l*-SL, *al*-SL and SL) are implemented using C language and executed on Intel Pentium 4 CPU (3.2 GHz) with 512 MB RAM PC. Nearest neighbor array [30] is used to implement SL method.

The *l*-SL and the *al*-SL methods are tested with Spiral dataset and results are reported in Table 2. The *l*-SL method produces five (5) clusters of arbitrary shapes (Fig. 3). However, it cannot find exact number of clusters (i.e. 2) of the data. On the other hand, the *al*-SL method finds exactly two arbitrary shaped clusters of the data (Fig. 4). Further, the *l*-SL method is significantly faster than

Table 1
Datasets used.

Dataset	# Patterns	# Features
DNA	2000	180
Banana	4900	2
Spiral (Synthetic)	3330	2
Pendigits	7494	16
Shuttle	58,000	9
GDS10	23,709	28
Circle4 (Synthetic)	28,000	2

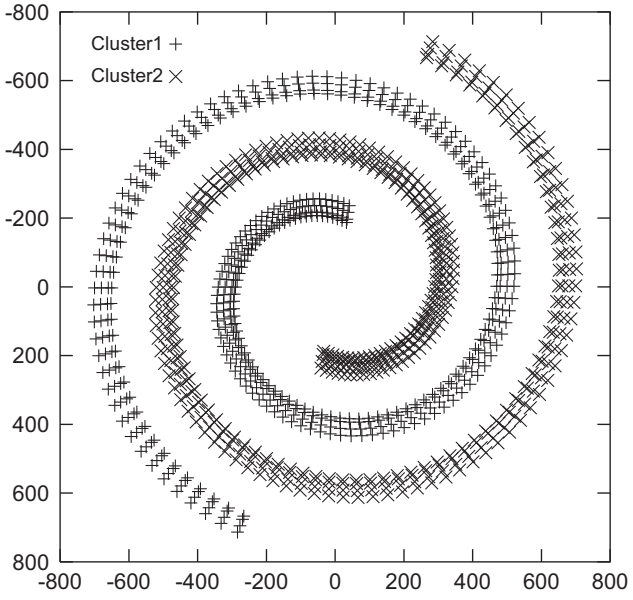

Fig. 2. Spiral dataset.

Table 2
Experimental results with standard datasets.

Dataset	Distance (h)	Method	Time (s)	Rand index (*RI*)
Spiral (Synthetic)	50.0	*l*-SL	0.08	0.808
	50.0	*al*-SL	0.12	1.000
	50.0	SL	20.85	–
	40.0	*l*-SL	0.12	1.000
	40.0	*al*-SL	0.14	1.000
	40.0	SL	19.32	–
DNA	11.00	*l*-SL	6.68	1.000
	11.00	*al*-SL	9.61	1.000
	11.00	SL	250.28	–
	9.00	*l*-SL	6.72	1.000
	9.00	*al*-SL	9.61	1.000
	9.00	SL	232.84	–
Banana	0.70	*l*-SL	0.01	0.999
	0.70	*al*-SL	0.04	1.000
	0.70	SL	43.55	–
	0.50	*l*-SL	0.02	0.998
	0.50	*al*-SL	0.05	1.000
	0.50	SL	42.60	–
	0.30	*l*-SL	0.07	0.997
	0.30	*al*-SL	0.11	1.000
	0.30	SL	41.68	–

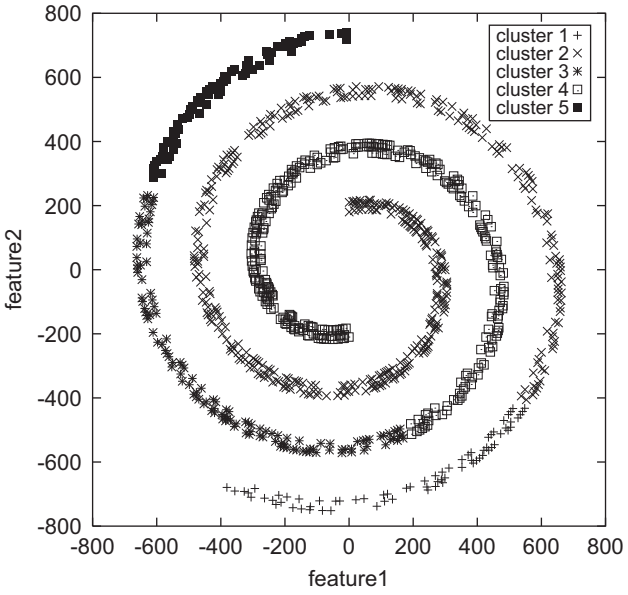

Fig. 3. Clusters obtained by *l*-SL method.

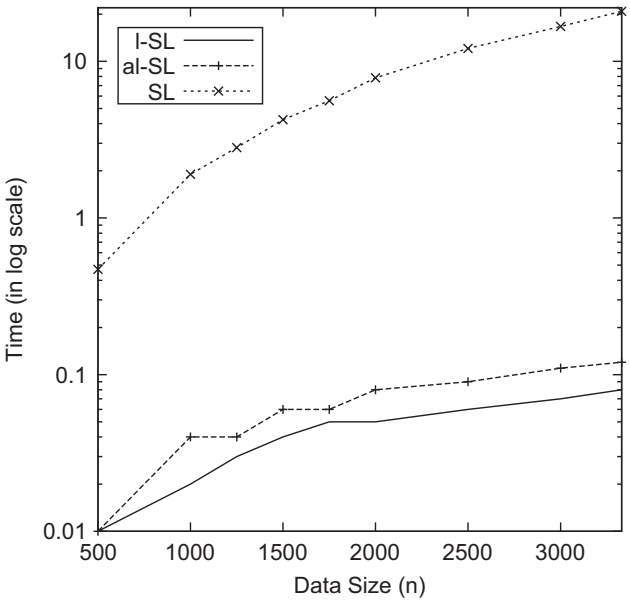

Fig. 5. Execution time of the *l*-SL, *al*-SL and SL methods for spiral dataset.

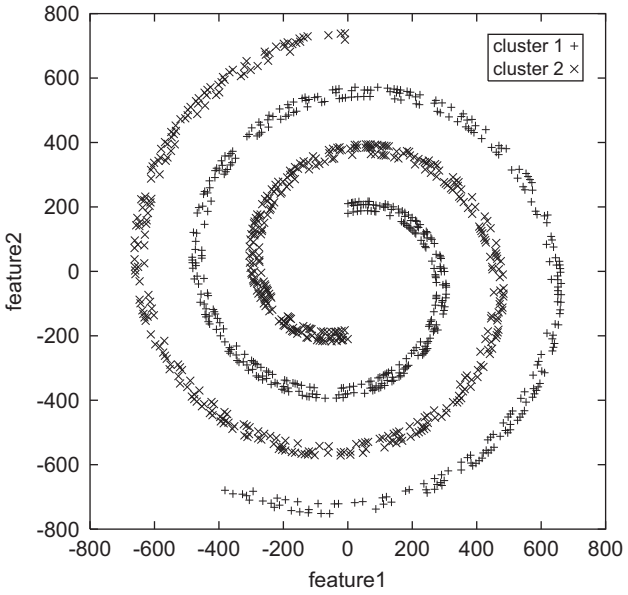

Fig. 4. Clusters obtained by *al*-SL method.

that of the SL method. Clustering results of the *l*-SL method ($RI=0.808$) is not same as that of the SL method ($RI=1.000$) (Table 2). The *al*-SL method produces same clustering results as produced by the SL method ($RI=1.000$). However, execution time of the *al*-SL method is slightly higher than the *l*-SL method.

For the DNA dataset, *l*-SL and *al*-SL both produce same clustering results ($RI=1.000$) as produced by the SL method. Both of these methods are significantly faster than that of the SL method.

In case of the Banana data, *l*-SL could not produce same results as produced by the SL method, but *l*-SL is significantly faster than that of the SL method. The *al*-SL method produces the same results as produced by the SL method. The *al*-SL method takes slightly more time compared to the *l*-SL method.

To show performance of the proposed methods with different dataset sizes, experiments are performed using Spiral data with different data sizes. Results are reported in Fig. 5 for $h=50$. This results show that the schemes are more scalable and effective for large datasets.

To show the time requirement of the merging process in *al*-SL method, experiments are also performed with the Spiral dataset. In the merging part of the *al*-SL method, potential leaders are searched for possible merging of clusters. The plot (Fig. 6) shows the percentage of total leaders to be searched with respect to distance (h). It shows that *al*-SL method selects only few number of leaders ($<10\%$) for possible merging in wide range of h values.

6.2. Experiment with large datasets

Experiments are also conducted with large datasets on Intel Xeon Processor (3.6 GHz) with 8 GB RAM IBM Server to show the effectiveness of the proposed clustering methods in large datasets. Experimental results are reported in Table 3.

The synthetic dataset Circle4 contains four (4) circles of same radius and they are well separated from each other. The results of the *l*-SL method are very close to the result of the SL method. *l*-SL method is significantly faster than SL method. *al*-SL method produces same clustering as that of SL method. However, its execution time is marginally higher than *l*-SL method. Similar results are also observed for other three real datasets (Pendigits, Shuttle and GDS10).

Fig. 7 shows execution time of the three methods for different dataset size. Dataset size varies from 5000 to 40,000 for $h=0.01$ with Shuttle dataset. When dataset size is more than 40,000, execution could not be completed due to memory shortage. To store distance matrix for the entire dataset, SL method consumes $58{,}000\times 58{,}000\times 4 = 12.54$ GB memory. Therefore, experimental results of SL method with whole dataset are not reported. Whereas, *l*-Sl and *al*-SL method produce results within 20 seconds for the whole dataset. We report Rand Index (RI) between the results of *l*-SL and the results of *al*-SL methods for this dataset (Table 3).

From GEO database (http://www.ncbi.nlm.nih.gov/geo/), we select a dataset named GDS10. Patterns with missing values are removed from this dataset. Finally, we use 23,709 out of 39,114 patterns for our experiments.

For all datasets, similar trends are found (as we discussed and proved theoretically), i.e. *l*-SL is fastest with approximate SL

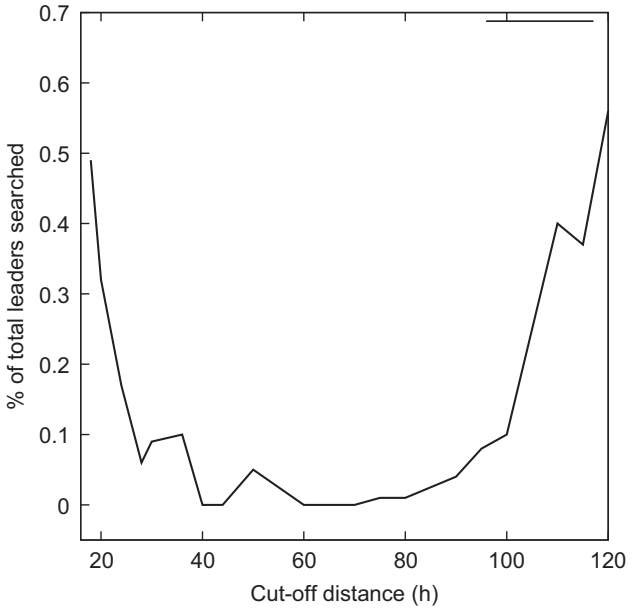

Fig. 6. The percentages of leaders searched by the *al*-SL method for spiral dataset.

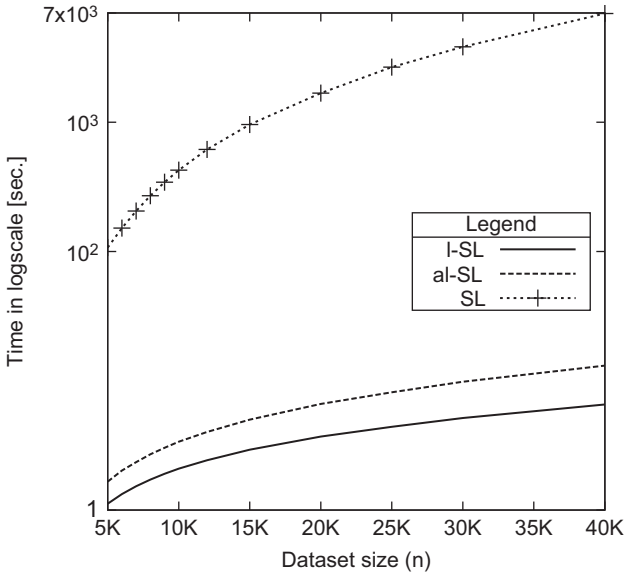

Fig. 7. Execution time of the *l*-SL, *al*-SL and SL method for shuttle dataset.

Table 3
Experimental results with large datasets.

Dataset	Distance (h)	Method	Time (s)	Rand index (RI)
Pendigits	90.0	*l*-SL	0.46	0.999
	90.0	*al*-SL	0.79	1.000
	90.0	SL	392.59	–
	70.0	*l*-SL	1.38	0.993
	70.0	*al*-SL	2.14	1.000
	70.0	SL	430.46	–
Shuttle	0.02	*l*-SL	9.13	0.999
	0.02	*al*-SL	19.38	–
	0.02	SL (40,000)	6929.77	
	0.01	*l*-SL	9.32	0.999
	0.01	*al*-SL	20.27	–
	0.01	SL (40,000)	6929.77	
GDS10	700	*l*-SL	0.50	0.999
	700	*al*-SL	1.15	1.000
	700	SL	4105.35	–
	900	*l*-SL	0.26	0.999
	900	*al*-SL	0.62	1.000
	900	SL	4105.35	–
Circle4 (Synthetic)	0.08	*l*-SL	19.65	0.995
	0.08	*al*-SL	36.02	1.000
	0.08	SL	948.26	–
	0.20	*l*-SL	19.35	1.000
	0.20	*al*-SL	20.6	1.000
	0.20	SL	960.25	–

clustering results. *al*-SL is slightly slower than *l*-SL method but significantly faster than SL method with exact SL clustering results.

Considering the experimental and theoretical results, it is clear that the proposed methods are significantly fast and suitable for large datasets.

7. Conclusions

The SL method can find arbitrary (non-convex) shaped clusters in a dataset. However, SL method scans dataset many times. It has time and space complexity of $O(n^2)$. So, SL method is not suitable for large datasets. In this paper, we proposed two clustering methods to detect arbitrary shaped clusters for large datasets. The first method, *l*-SL takes considerably less time compared to that of the SL method and scans the dataset only once. Clustering results produced by *l*-SL is very close to that of the SL method. So, *l*-SL is suitable for the applications where faster and approximate clustering suffice.

The second method, *al*-SL produces exact clustering results as produced by the SL method. Execution time of the *al*-SL method is slightly more than that of the *l*-SL method. Experimental results confirm that proposed methods are fast and suitable for large datasets.

Acknowledgments

The authors are thankful to Prof. Vijay S. Iyengar, Fellow, IEEE and Dr. Sanasam Ranbir Singh, IIT Guwahti for their valuable suggestions. This work is supported by Council of Scientific & Industrial Research (CSIR), Government of India.

References

[1] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., Wiley Interscience Publication, New York, 2000.
[2] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Computing Surveys 31 (3) (1999) 264–323.
[3] A.K. Jain, R.P. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (1) (2000) 4–37.
[4] R. Xu, D. Wunsch, Survey of clustering algorithms, IEEE Transactions on Neural Networks 16 (3) (2005) 645–678.
[5] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability1967, pp. 281–297.
[6] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, John Wiley & Sons, USA, 1990.
[7] R.T. Ng, J. Han, CLARANS: a method for clustering objects for spatial data mining, IEEE Transactions on Knowledge and Data Engineering 14 (5) (2002) 1003–1016.
[8] M. Ester, H.P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)1996, pp. 226–231.
[9] A. Hinneburg, D.A. Keim, An efficient approach to clustering in large multimedia databases with noise, in: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)1998, pp. 58–65.

[10] P.H.A. Sneath, R.R. Sokal, Numerical Taxonomy, Freeman, London, 1973.
[11] B. King, Step-wise clustering procedures, Journal of the American Statistical Association 62 (317) (1967) 86–101.
[12] M.H. Dunham, Data Mining: Introductory and Advanced Topics, Prentice-Hall, New Delhi, 2003.
[13] M. Ankerst, M.M. Breunig, H.-P. Kriegel, J. Sander, Optics: ordering points to identify the clustering structure, SIGMOD Record 28 (2) (1999) 49–60.
[14] G. Karypis, E.-H.S. Han, V. Kumar, Chameleon: hierarchical clustering using dynamic modeling, Computer 32 (8) (1999) 68–75.
[15] J. Kleinberg, An impossibility theorem for clustering, in: Proceeding of the 16th Conference on Neural Information Processing Systems (NIPS)2002, pp. 446–453.
[16] A. Krause, J. Stoye, M. Vingron, Large scale hierarchical clustering of protein sequences, BMC Bioinformatics 6 (15) (2005).
[17] M. Dash, H. Liu, P. Scheuermann, K.L. Tan, Fast hierarchical clustering and its validation, Data & Knowledge Engineering 44 (1) (2003) 109–138.
[18] M. Nanni, Speeding-up hierarchical agglomerative clustering in presence of expensive metrics, in: Proceedings of Ninth Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)2005, pp. 378–387.
[19] P.A. Vijaya, M.N. Murty, D.K. Subramanian, Efficient bottom-up hybrid hierarchical clustering techniques for protein sequence classification, Pattern Recognition 39 (12) (2006) 2344–2355.
[20] H. Koga, T. Ishibashi, T. Watanabe, Fast agglomerative hierarchical clustering algorithm using Locality-Sensitive Hashing, Knowledge and Information Systems 12 (1) (2007) 25–53.
[21] P. Indyk, R. Motwani, Approximate nearest neighbors:towards removing the curse of dimensionality, in: Proceedings of 30th ACM Symposium on theory of Computing1998, pp. 604–613.
[22] M.N. Murty, G. Krishna, A hybrid clustering procedure for concentric and chain-like clusters, International Journal on Computer Information Science 10 (6) (1981) 397–412.
[23] M.A. Wong, A hybrid clustering algorithm for identifying high density clusters, Journal of the American Statistical Association 77 (380) (1982) 841–847.
[24] C.-R. Lin, M.-S. Chen, Combining partitional and hierarchical algorithms for robust and efficient data clustering with cohesion self-merging, IEEE Transactions on Knowledge and Data Engineering 17 (2) (2005) 145–159.
[25] M. Liu, X. Jiang, A.C. Kot, A multi-prototype clustering algorithm, Pattern Recognition 42 (2009) 689–698.
[26] J.A. Hartigan, Clustering Algorithms, John Wiley & Sons Inc., New York, 1975.
[27] H. Spath, Cluster Analysis Algorithms for Data Reduction and Classification of Objects, Ellis Horwood, UK, 1980.
[28] B.K. Patra, S. Nandi, A Fast Single Link Clustering Method Based on Tolerance Rough Set Model, in: Proceedings of 12th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC)2009, pp. 414–422.
[29] P. Viswanath, V. Babu, Rough-DBSCAN: a fast hybrid density based clustering method for latge data sets, Pattern Recognition Letters 30 (16) (2009) 1477–1488.
[30] C.F. Olson, Parallel algorithms for hierarchical clustering, Parallel Computing 21 (1995) 1313–1325.
[31] S. Theodoridis, K. Koutroumbas, Pattern Recognition, third ed., Academic Press Inc., Orlando, 2006.
[32] A. Juan, E. Vidal, Comparison of four initialization techniques for the *k*-medians clustering algorithm, in: Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition2000, pp. 842–852.
[33] J.P. Tremblay, R. Manohar, Discreate Mathematical Structures with Applications to Computer Science, Tata McGraw-Hill Publishing Company Limited, New Delhi, 1997.
[34] A.K. Jain, M.H.C. Law, Data clustering: a user's dilemma, in: Proceedings of First International Conference on Pattern Recognition and Machine Intelligence (PReMI)2005, pp. 1–10.
[35] W.M. Rand, Objective Criteria for Evaluation of Clustering Methods, Journal of American Statistical Association 66 (336) (1971) 846–850.

Bidyut Kr. Patra received B.Sc. (Physics), B.Tech. (CSE) and M.Tech. (CSE) from Calcutta University, Kolkata, India in 1996, 1999 and 2001, respectively. He has been working for his Ph.D. at the Indian Institute of Technology-Guwahati, India since July, 2005. He worked as a Lecturer at Haldia Institute of Technology, Haldia, West Bengal, India during August, 2001-June, 2005. Currently, he is an Assistant Professor at Tezpur Central University, Tezpur, Assam, India. His research interests include Pattern Recognition, Data Mining and Algorithms.

Sukumar Nandi received B.Sc. (Physics), B Tech and M Tech from Calcutta University in 1984, 1987 and 1989, respectively. He received the Ph. D. degree in Computer Science and Engineering from Indian Institute of Technology Kharagpur in 1995. In 1989-90 he was a faculty in Birla Institute of Technology, Mesra, Ranchi, India. During 1991 to 1995, he was a scientific officer in Computer Sc & Engg, Indian Institute of Technology Kharagpur. In 1995 he joined in Indian Institute of Technology Guwahati as an Assistant Professor in Computer Science and Engineering. Subsequently, he became Associate Professor in 1998 and Professor in 2002. Presently, he is the Dean of Academic Affairs of Indian Institute of Technology Guwahati. He was with the School of Computer Engineering, Nanyang Technological University, Singapore as Visiting Senior Fellow for one year (2002-2003). He was member of Board of Governor, Indian Institute of Technology Guwahati for 2005 and 2006. He was General Vice-Chair of 8th International Conference on Distributed Computing and Networking (ICDCN) 2006. He was General Co-Chair of the 15th International Conference on Advance Computing and Communication (ADCOM) 2007. He is also involved in several international conferences as member of advisory board/ Technical Programme Committee. He is reviewer of several international journals and conferences. He is co-author of a book titled Theory and Application of Cellular Automata published by IEEE Computer Society. He has published more than 150 Journals/Conferences papers. His research interests are Computer Networks (Traffic Engineering, Wireless Networks), Computer and Network security, Data mining. He is Senior Member of IEEE, Senior Member of ACM, Fellow of the Institution of Electronics and Telecommunication Engineers and Fellow of the Institution of Engineers (India).

P. Viswanth received his M.Tech (CSE) from the Indian Institute of Technology-Madras, Chennai, India in 1996. From 1996 to 2001, he worked as a faculty member at BITS-Pilani, India and Jawaharlal Nehru Technological University, Hyderabad, India. He received his Ph.D. from the Indian Institute of Science, Bangalore, India in 2005. He received Alumni Medal from IISc-Bangalore as his Ph.D. thesis was selected as the best thesis in the Electrical Sciences Division of IISc-Bangalore in the year 2006. He worked as Assistant Professor in the Department of Computer Science and Engineering, IIT-Guwahati, Guwahati, India from February 2005 to July 2008. At present he is working as a Professor and Dean, R&D (Electrical Sciences) at Rajeev Gandhi Memorial College of Engineering & Technology, Nandyal, Andhra Pradesh, India. His areas of interest include Pattern Recognition, Data Mining, Databases and Algorithms.