Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition Letters 25 (2004) 1293–1302
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Cluster center initialization algorithm for *K*-means clustering

Shehroz S. Khan [a,*], Amir Ahmad [b]

[a] *Scientific Analysis Group, DRDO, Metcalfe House, Delhi 110054, India*
[b] *Solid State Physics Laboratory, DRDO, Probyn Road, Delhi 110054, India*

Received 13 May 2003; received in revised form 16 March 2004
Available online 8 June 2004

Abstract

Performance of iterative clustering algorithms which converges to numerous local minima depend highly on initial cluster centers. Generally initial cluster centers are selected randomly. In this paper we propose an algorithm to compute initial cluster centers for *K*-means clustering. This algorithm is based on two observations that some of the patterns are very similar to each other and that is why they have same cluster membership irrespective to the choice of initial cluster centers. Also, an individual attribute may provide some information about initial cluster center. The initial cluster centers computed using this methodology are found to be very close to the desired cluster centers, for iterative clustering algorithms. This procedure is applicable to clustering algorithms for continuous data. We demonstrate the application of proposed algorithm to *K*-means clustering algorithm. The experimental results show improved and consistent solutions using the proposed algorithm.

Keywords: *K*-Means clustering; Initial cluster centers; Cost function; Density based multiscale data condensation

1. Introduction

Partitioning a given set of points into homogeneous groups is one of the most fundamental problem in pattern recognition. Clustering is one of the widely used knowledge discovery technique to reveal structures in a data set that can be extremely useful to the analyst. Clustering has a variety of applications in different domains viz data mining and knowledge discovery (Fayyad et al., 1996), data compression and vector quantization (Gersho and Gray, 1992), optimization (Bradley et al., 1997), finance, manufacturing and medical organizations. The problem of clustering is to partition a data set consisting of n points embedded in m-dimensional space into K distinct set of clusters such that the data points within a cluster are more 'similar' among them than to data points in other cluster. The term *similar*, when applied to clusters, means closer by some similarity measure. There are a number of proximity indices that have been used as similarity measures (Anderberg, 1973).

K-Means clustering algorithm (Mac Queen, 1967), developed three decades ago is one of the

* Corresponding author. Tel.: +91-1123955345; fax: +91-1123919828.
E-mail address: shehrozkhan@rediffmail.com (S.S. Khan).

0167-8655/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2004.04.007

most popular clustering algorithm used in variety of domains. A priori knowledge of number of clusters are must for K-means clustering algorithm. K-Means is defined over continuous data (Fukunaga, 1990) (Duda and Hart, 1973). K-Means algorithm calculates its centers iteratively (Gersho and Gray, 1992). Let $D = \{d_i \,|\, i = 1, \ldots, n\}$ be a data set having K-clusters, $C = \{c_i \,|\, i = 1, \ldots, K\}$ be a set of K centers and $S_j = \{d \,|\, d$ is member of cluster $k\}$ be the set of samples that belong to the kth cluster. K-Means algorithm minimizes the following function which is defined as a cost function

$$\text{Cost} = \sum_{i=1}^{n} \text{dist}(d_i, c_k) \tag{1}$$

where $\text{dist}(d_i, c_k)$ measures the Euclidean distance between a pattern d_i and its cluster center c_k. The K-means algorithm calculate cluster centers iteratively as follows:

1. Initialize the centers in c_k using random sampling
2. Decide membership of the patterns in one of the K-clusters according to the minimum distance from cluster center criteria
3. Calculate new c_k centers as:

   $$c_k = \frac{\sum_{d_i \in S_k} d_i}{|S_k|}$$

   $|S_k|$ is the number of data items in the kth cluster
4. Repeat steps 2 and 3 till there is no change in cluster centers.

K-Means does not guarantee unique clustering because we get different results with randomly chosen initial clusters. Machine learning practitioners find it difficult to rely on the results thus obtained. The K-means algorithm gave better results only when the initial partitions was close to the final solution (Jain and Dubes, 1988). Several attempts have been reported to solve the cluster initialization problem. A recursive method for initializing the means by running K clustering problems is discussed by Duda and Hart (1973). A variant of this method consists of taking the mean of the entire data and then randomly perturbing it K times (Thiesson et al., 1997). Bradley et al. (1997) reported that the values of initial means along any one of the m coordinate axes is determined by selecting the K densest "bins" along that coordinate. Bradley and Fayyad (1998) proposes a procedure that refines the initial point to a point likely to be close to the modes of the joint probability density of the data. Penã et al. (1999) presented a comparative study for different initialization methods for the K-means algorithm. The result of their experiments illustrate that the random and the Kaufman initialization method outperforms the rest of the compared methods as they make the K-means more effective and more independent on initial clustering and on instance order.

In Section 2, we present our proposed cluster center initialization algorithm (CCIA) and an algorithm based on density based multiscale data condensation (Mitra et al., 2002) to merge similar clusters. Section 3 shows the experimental run of CCIA on Fossil data (Yi-tzuu, 1978) and demonstrate improved and consistent clustering in comparison to random initialization. In Section 4, we present some simulation results on real world data sets. Conclusion follows in Section 5.

2. Cluster center initialization algorithm (CCIA)

In iterative clustering algorithms the procedure adopted for choosing initial cluster centers is extremely important as it has a direct impact on the formation of final clusters. Since clusters are separated groups in a feature space, it is desirable to select initial centers which are well separated. It is dangerous to select outliers as initial centers, since they are away from normal samples. Our proposed algorithm calculates the initial cluster centers that are quite close to the desired cluster centers. As there are no universally accepted method for selecting initial cluster center as reported by Meila and Heckerman (1998), we compare the results against the standard method of randomly choosing initial starting points.

We observed that if data sets are clustered repetitively with random initialization using

K-means clustering algorithm, some of the patterns have same cluster membership consistently. In other words they belong to same clusters irrespective of initialization. For example $D = \{d_1, \ldots, d_n\}$ be a dataset consisting of n patterns. Let us assume d_{m1}, d_{m2}, d_{m3}, d_{m4}, where $1 \leqslant m1, m2, m3, m4 \leqslant n$ are very similar then they have same cluster membership whenever K-means algorithm is executed using different initial points. This information can be used to compute initial centers.

Instead of using random initialization every time, we follow a novel approach to provide initial cluster centers. This approach is based on the experimental fact that individual attribute can provide some lead to initial cluster centers. The first step of the proposed algorithm is the computation of cluster centers for individual attributes. To achieve this we use K-means algorithm over this attribute. To instigate K-means algorithm we need some initial centers. Instead of providing random cluster centers we provide centers that are far apart keeping the constraint that outliers are removed.

We assume that each of the attributes of the pattern space are *normally distributed*. For K-fixed clusters we divide the normal curve into K partitions such that the area under these partitions is equal. We then take the midpoints of the interval of each of these partitions. This has been done to scrap the outliers and to keep the centers as far as possible. We calculate the percentile (Neter et al., 1992) of the sth point such that the area from $-\infty$ to sth mid-point is equal to $\frac{2s-1}{2K}$, $s = 1, 2, \ldots, K$. We compute the attribute values corresponding to these percentiles using mean and standard deviation of the attribute. This will serve as seed point for the K-means clustering for this attribute. It is possible that this attribute may not follow normal distribution. But by running K-means algorithm over this attribute, it is more likely that we get centers where the density of attribute values is high, hence these centers are good representative of clusters. As this process accomplishes, every attribute values of this attribute are associated with some clusters. Now we run K-means algorithm on entire data set with initial cluster membership achieved by above mentioned process. We repeat the whole process for all the attributes. After this process we get a sequence of m cluster labels for every pattern. We call it as a *pattern string*. These classes of pattern string may or may not be same. For n patterns we will have n such pattern strings, where ith entry of the jth pattern string corresponds to the class of the jth pattern when the initial centers were based on ith attribute.

Suppose we obtain K' distinct pattern strings. These pattern strings represent K' clusters. We compute the cluster centers for these K' clusters. If K' is equal to K, then centers of these K' clusters should be treated as the initial cluster centers for the K-means algorithm. If K' is greater then the number of desired clusters (K), we merge similar clusters to get K-clusters and centers of these K-clusters will become initial cluster centers for the K-means algorithm. The merging of clusters is achieved by using density-based multiscale data condensation method which is briefly discussed in the next subsection.

2.1. Density-based multi scale data condensation

A data set can be replaced by a subset of representative patterns such that the accuracy of estimates (e.g. of probability density, dependencies, class boundaries) obtained from such a reduced set should be comparable to that obtained using the entire data set. There exists various approaches for data reduction such as random sampling, stratified sampling and peepholing (Catlett, 1991), uncertainty sampling (Lewis and Catlett, 1994), and active learning (Roy and McCallum, 2001). Mitra et al. (2002) proposed a density-based multi scale data condensation (DBMSDC) method for selecting a small representative subset from a data set. They experimentally showed the superiority of their approach as compared to several related condensation methods both in terms of condensation ratio and estimation error. This data condensation algorithm involves estimating the density at a point, sorting the points based on the density criterion, selecting a point according to the sorted list and pruning all points lying within a disc about a selected point with radius inversely proportional to the density at that point.

CCIA generates K' clusters which may be greater than the desired number of clusters K. In this situation our aim is to merge some of the similar clusters so as to get K clusters. Since the clusters are well represented by their centers, we can merge those clusters whose centers are near to each other. For merging similar clusters, the center of each cluster is computed and K-representative subset of centers are selected from the K' cluster centers using DBMSDC algorithm. These K-subsets provide us with the information that which of these clusters are alike. It means clusters with cluster center in the same subset are similar and will be merged. For every K-subset, the clusters contained in them are merged and center of merged clusters are computed and treated as initial cluster center of the K-means algorithm. Since we have K-subsets we will have K-initial cluster centers.

2.2. Algorithm (CCIA)

In this subsection we present execution steps of our proposed cluster center initialization algorithm (CCIA) for K-means clustering. This algorithm consists of two parts. The first part deals with generation of K' cluster centers. If $K' > K$ then we execute the second part of the algorithm to merge similar clusters to get K cluster centers. These K cluster centers are taken as initial cluster center for K-means algorithm.

Input:
D—the set of n data elements described with attributes $A_1, A_2, \ldots, A_m$ where $m =$ no. of attributes and all attributes are numeric
K—predefined number of clusters
Output:
Initial cluster centers for K-means algorithm

1. For each attribute A_j repeat steps 2–9
2. Compute mean (μ_j) and standard deviation (σ_j)
3. Compute percentile z_s, corresponding to area under the standard normal curve from $-\infty$ to z_s equals to $\frac{2s-1}{2K}$, where $s = 1, 2, \ldots, K$
4. Compute attribute value corresponding to these percentiles using means and standard deviation of the attribute as $x_s = z_s * \sigma_j + \mu_j$
5. Create initial partitions using Euclidean distance between x_s and A_j^{th} attribute of all patterns (The assigned class label is treated as the class of the pattern)
6. Execute K-means on this attribute
7. Allocate cluster labels obtained from step 6 to every pattern and compute new dense centers
8. Execute K-means on complete data set
9. Store the class labels as S_{tj} where $t = 1, 2, \ldots, n$
10. Generate pattern string, P_t corresponding to every pattern by storing the class labels. Every pattern string will have m class labels
11. Find unique strings, K', which is the number of distinguishable clusters and $K' \geqslant K$. Find the center of each of these clusters
12. If $K' > K$, apply Algorithm *Merge*-DBMSDC to these K' centers to get K-set of points
13. Merge those clusters whose centers are occurring in the same set. Since we have K-set of points we get K-clusters
14. Find the mean of these K-clusters to get final K-centers that will be used as initial cluster centers

The computation of K-subsets from K' cluster centers using DBMSDC is described as under:

Algorithm *Merge*DBMSDC

Input:
K' $(> K)$ cluster centers
Output:
K cluster centers

1. Let K' is the number of clusters generated by CCIA and $K' > K$
2. Compute cluster center for every K' cluster
3. Let $B = \{x_1, x_2, \ldots, x_{K'}\}$ be the set of K' cluster centers
4. Choose a positive integer q and initialize $l = 1$ and repeat steps 5–10 till $B = \phi$
5. For each cluster center $x_i \in B$, calculate the distance of the qth nearest neighbor of x_i in B. Denote it by r_{q,x_i}

6. Select the point $x_j \in B$, having the lowest value of r_{q,x_j}. Ties in the lowest value of r_{q,x_i} may be resolved by following some convention like the index of the samples etc.
7. Create a set $S_l = \phi$
8. Add x_j to S_l
9. Remove all points from B that lie with in a disc of radius $1.5r_{q,x_j}$ centered at x_j and add them to S_l. The set B consisting of the remaining centers is to be renamed as B.
10. Increment l by 1 i.e. $l = l + 1$;

We choose q nearest neighbors such that steps 1–10 is repeated K-times. After this process we will have K-subsets of similar cluster centers.

3. Experimental run of CCIA

The purpose of this experiment is to show how close the initial cluster centers computed by proposed algorithm CCIA, are to the desired cluster centers. We present the complete experimental run of CCIA on Fossil data taken from Chernoff (Yi-tzuu, 1978). It consists of 87 nummulitidae speciments from the Eocene yellow limestone formation of north western Jamaica. Each specimen is characterized by six measurements. There are three cluster groups as identified by Chernoff, which contains 40(I), 34(II) and 13(III) patterns each. If we run the K-means algorithm, most of the time it merges clusters 1 and 3. The computational steps for calculating the initial cluster centers using CCIA are as under:

1. Normalize the complete data set (as discussed in Section 4)
2. For attribute A_1 repeat steps 3–10
3. Compute mean (μ_1) and standard deviation (σ_1), $\mu_1 = 0.391$, $\sigma_1 = 0.272$
4. Since this is a three class problem therefore $K = 3$. Compute the percentiles z_1, z_2, z_3 such that the area under the normal curve from
   $-\infty$ to z_1 is equal to 1/6,
   $-\infty$ to z_2 is equal to 1/2 and
   $-\infty$ to $z_3 = 5/6$

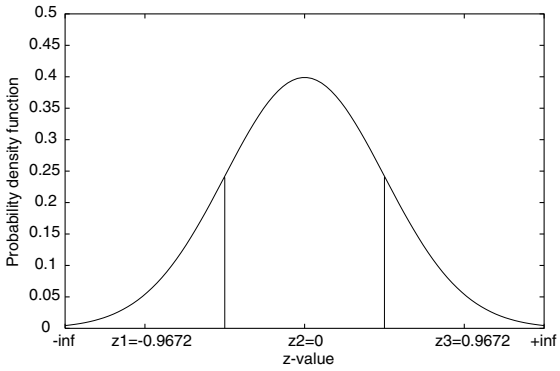

Fig. 1. Computation of percentiles z_s for attribute A_1 for $K = 3$ (step 3 of CCIA).

Therefore, $z_1 = -0.9672$, $z_2 = 0.0$, $z_3 = 0.9672$ (Fig. 1)

5. Compute attribute values corresponding to these percentiles

$$x_s = z_s * \sigma_1 + \mu_1, \quad s = 1, 2, 3$$
$$x_1 = 0.127, \quad x_2 = 0.391, \quad x_3 = 0.655$$

6. Create initial partitions using Euclidean distance between x_s and this attribute of all patterns
7. Execute K-means over this attribute
8. Allocate cluster labels obtained from step 7 to every pattern and compute the new dense centers as 0.164, 0.497, 0.860 (Fig. 2)
9. Execute K-means on complete data set
10. Store the class labels
11. Repeat the above procedure for the remaining five attributes to arrive at 87 different *pattern strings* such that every pattern gets associated with six class labels e.g.
    Pattern 1—121312
    Pattern 2—121312
    Pattern 34—121312
    Pattern 41—212123
    Pattern 42—312123
    Pattern 50—212123
    Pattern 51—312123
    Pattern 65—212123
    Pattern 76—133231
    Pattern 87—133231 etc.
12. Find the unique strings (K' clusters), that is
    string 121312 (cluster 1) has count 40
    string 212123 (cluster 2) has count 20

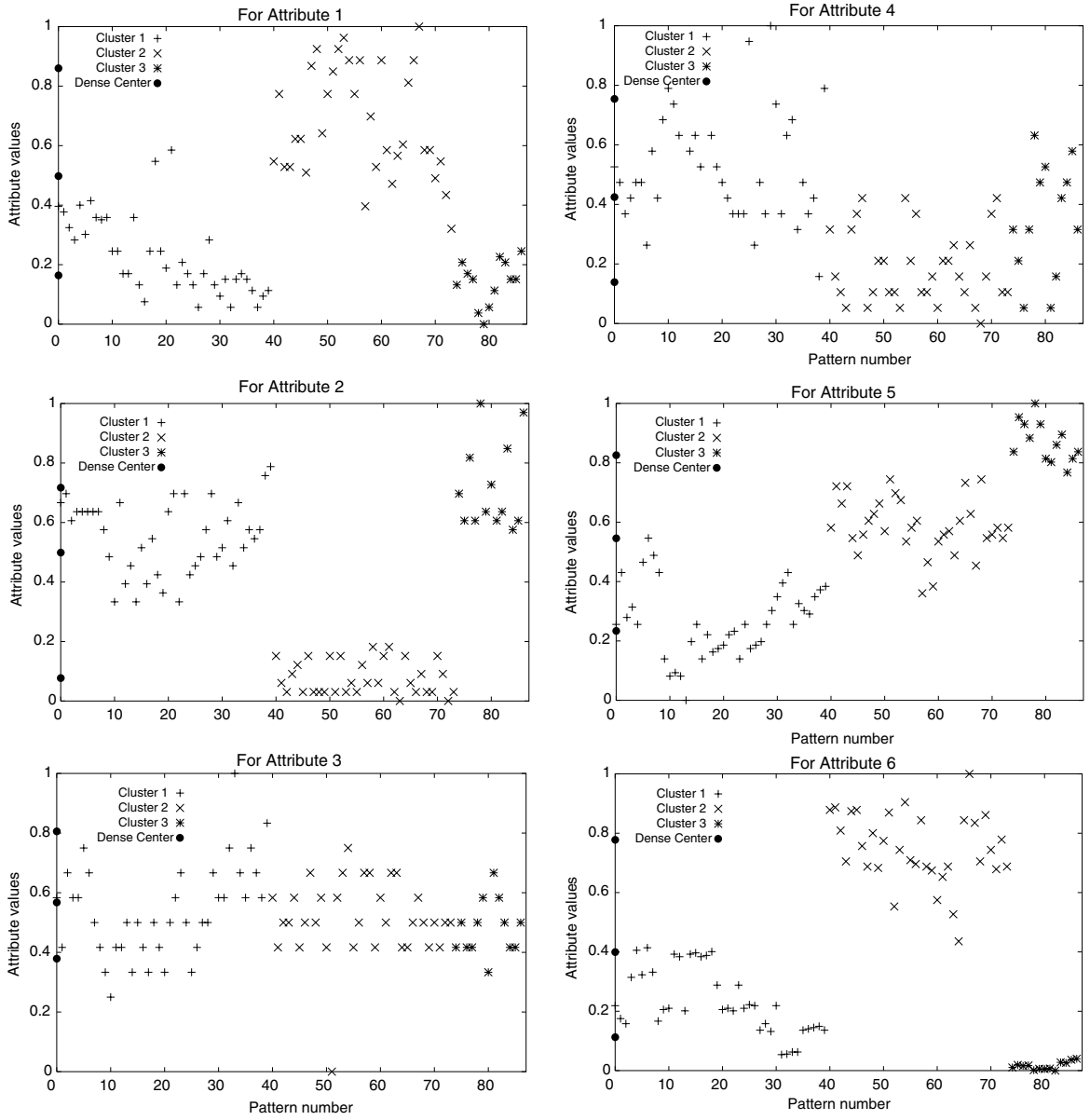

Fig. 2. Values of centers of each attribute where density of attribute values in quite high.

string 312123 (cluster 3) has count 14
string 133231 (cluster 4) has count 13
and hence $K' = 4$. Find the center of each of these distinguishable clusters i.e.
Center of cluster 1, 0.230, 0.553, 0.539, 0.538, 0.265, 0.232
Center of cluster 2, 0.540, 0.080, 0.525, 0.207, 0.560, 0.727
Center of cluster 3, 0.872, 0.073, 0.500, 0.161, 0.622, 0.777
Center of cluster 4, 0.142, 0.717, 0.480, 0.348, 0.871, 0.016

13. Since $K' > K$, apply *Merge*DBMSDC algorithm. to get three set of points → {1}, {2,3}, {4}
14. Clusters 2 and 3 are merged because they belong to same set and we are left with three clusters
15. Find the mean of these three clusters to get final three centers that are the initial cluster centers for K-means algorithm
    0.230, 0.553, 0.540, 0.538, 0.265, 0.232
    0.677, 0.078, 0.515, 0.189, 0.586, 0.748
    0.142, 0.718, 0.481, 0.348, 0.871, 0.016.

4. Results and discussion

To establish practical applicability of the CCIA algorithm, we implemented it and tested its performance on a number of other real world data sets, the fossil data, the wine recognition data, the Ruspini data and the letter image recognition data.

Since different attributes are measured on different scales, when Euclidean distance formula is used directly, the effect of some attributes might be completely dwarfed by others that have larger scales of measurement. Consequently it is usual to normalize all attribute values to lie between 0 and 1 (Witten and Frank, 2000).

The data sets used for the evaluation include a "correct answer" or label for each pattern. We use the labels in a post processing step for evaluating performance. The error that we have calculated depends on number of misclassified patterns and the total number of patterns in the dataset.

$$\text{Error(in \%age)} = \frac{\text{Number of misclassified patterns}}{\text{Total number of patterns}} \times 100$$

For computing error for K-means algorithms with random initial cluster centers, K-means was executed 100 times and the average error is taken as the performance measure.

To measure the degree of closeness between the initial cluster centers and the desired cluster centers we have defined the *Cluster Center Proximity Index* (*CCPI*) as

Table 1
Comparing *CCPI* of data sets

Data set	*CCPI*	
	CCIA	Random
Fossil data	0.0021	0.3537
Iris data	0.0396	0.8909
Wine data	0.1869	0.3557
Ruspini data	0.0361	1.2274
Letter image recognition data	0.0608	0.1572

$$CCPI = \frac{1}{K * m} \sum_{s=1}^{K} \sum_{j=1}^{m} \left| \frac{f_{sj} - C_{sj}}{f_{sj}} \right|$$

where f_{sj} is the jth attribute value of the desired sth cluster center and C_{sj} is the jth attribute value of the initial sth cluster center.

4.1. Iris data

This data set (Fisher, 1936) has often been used as a standard for testing clustering algorithms. This data set has three classes that represents three different varieties of Iris flowers namely Iris setosa(I), Iris versicolor(II) and Iris virginica(III). Fifty samples were obtained from each of three classes, thus a total of 150 samples is available. Every sample is described by a set of four attributes viz sepal length, sepal width, petal length and petal width. In numerical representation, two of the classes (virginica, versicolor) have a large overlap while third is well separated from the other two.

4.2. Wine recognition data

This data set is taken from UCI repository website (see references). This data set is the result of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines. There were overall 178 instances. There are 59, 71 and 48 instances in class I, class II and class III respectively. The classes are separable.

4.3. Ruspini data

The Ruspini data set (Ruspini, 1970) is popular to illustrate clustering techniques. It consists of

Table 2
Fossil data

Clusters found	Points in cluster	Coming from			Error with K-means using initial centers computed by CCIA	Average error with K-means using andom initialization
		I	II	III		
C1	40	40	0	0	0.00%	12.41%
C2	34	0	34	0		
C3	13	0	0	13		

Table 3
Fisher's Iris data

Clusters found	Points in cluster	Coming from			Error with K-means using initial centers computed by CCIA	Average error with K-means using random initialization
		I	II	III		
C1	50	50	0	0	11.33%	23.6%
C2	61	0	47	14		
C3	39	0	3	36		

Table 4
Wine data

Clusters found	Points in cluster	Coming from			Error with K-means using initial centers computed by CCIA	Average error with K-means using random initialization
		I	II	III		
C1	65	59	6	0	5.05%	5.61%
C2	62	0	62	0		
C3	51	0	3	48		

Table 5
Ruspini data

Clusters found	Points in cluster	Coming from				Error with K-means using initial centers computed by CCIA	Average error with K-means using random initialization
		I	II	III	IV		
C1	20	20	0	0	0	4.0%	8.8%
C2	23	3	20	0	0		
C3	17	0	0	17	0		
C4	15	0	0	0	15		

Table 6
Letter image recognition data

Clusters found	Points in cluster	Coming from		Error with K-means using initial centers computed by CCIA	Average error with K-means using random initialization
		A	D		
C1	551	522	29	8.55%	8.47%
C2	641	73	568		

75 observations on two variables making up four natural groups including 23, 20, 17 and 15 entities in classes I, II, III and IV respectively.

4.4. Letter image recognition data

This data set is also obtained from UCI repository website (see references). The objective is to

identify each of a large number of black-and-white rectangular pixel displays as one of the 26 capital letters in the English alphabet. The character images were based on 20 different fonts and each letter within these 20 fonts was randomly distorted to produce a file of 20,000 unique stimuli. Each stimulus was converted into 16 primitive numerical attributes (statistical moments and edge counts) which were then scaled to fit into a range of integer values from 0 through 15. The training data consists of first 16,000 items and testing data comprises of remaining 4000 items to predict the letter category. For experimental purpose we have taken 595 patterns of letter A and 597 patterns of letter D from the training data set.

The comparison of initial cluster centers computed using CCIA and desired cluster centers, for these data sets, is shown in Table 1. The clustering results using K-means clustering algorithm with initial cluster centers generated using CCIA and random initialization are presented (for these data sets) from Tables 2–6. The *CCPI* (Cluster Center Proximity Index) values presented in Table 1 show smaller values of *CCPI* when initial cluster centers were computed using CCIA in comparison to random initial center selection, for all data sets. Low value of *CCPI* is the measure of good selection of initial cluster centers.

Clustering results, thus obtained, with K-means algorithm using the initial centers computed by CCIA suggest that we get improved and consistent clusters for all data set in comparison to random initialization. We are getting better clustering results with K-means clustering algorithm using initial cluster centers computed by CCIA because of good selection of initial centers, which are very near to the desired cluster centers. This choice of initial cluster centers may avoid the K-means clustering algorithm to get trapped in one of the numerous local minima (Jain and Dubes, 1988).

5. Conclusion

We have presented an algorithm (CCIA) for computing initial cluster centers for iterative clustering algorithm. This procedure is based on the experimental fact that very similar data points (patterns) form the core of clusters and their cluster membership remain the same. However, the outliers are more susceptible to a change in cluster membership. Hence these similar patterns (which forms the core of clusters) aid in finding initial cluster centers. We observed that individual attribute also provide information in computing initial cluster centers. CCIA generate clusters which may be more than the number of desired clusters. Similar clusters are merged using density-based multiscale data condensation method to get the desired number of clusters. Center of these clusters have been used as initial clusters for the K-means clustering algorithm. Experimental results show improved and consistent cluster structures as compared to the random choice of initial cluster centers.

References

Anderberg, M.R., 1973. Cluster Analysis for Applications. Academic Press Inc.

Bradley, P.S., Fayyad, U.M., 1998. Refining initial points for K-means clustering. In: Sharlik, J. (Ed.), Proc. 15th Internat. Conf. on Machine Learning (ICML'98). Morgan Kaufmann, San Francisco, CA, pp. 91–99.

Bradley, P.S., Mangasarian, O.L., Street, W.N., 1997. Clustering via concave minimization. In: Mozer, M.C., Jordan, M.I., Petsche, T. (Eds.), Advances in Neural Information Processing System, vol. 9. MIT Press, pp. 368–374.

Catlett, J., 1991. Megainduction: Machine learning on very large database. Ph.D. Thesis, Department of Computer Science, University of Sydney, Australia.

Duda, R.O., Hart, P.E., 1973. Pattern Classification and Scene analysis. John Wiley and Sons, NY.

Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R., 1996. Advances in Knowledge Discovery and Data Mining. AAAI Press.

Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems. Ann. Eugenic. 7 (part 2), 179–188.

Fukunaga, K., 1990. Introduction to Statistical Pattern Recognition. Academic Press, San Diego, CA.

Gersho, A., Gray, R.M., 1992. Vector Quantization and Signal Compression. KAP.

Jain, A.K., Dubes, R.C., 1988. Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs.

Lewis, D.D., Catlett, J., 1994. Heterogeneous uncertainty sampling for supervised learning. In: Proc. 11th Internat. Conf. on Machine Learning, pp. 148–156.

Mac Queen, J., 1967. Some methods for classification and analysis of multivariate observations (pp. 281–297). In: Le Cam,

L.M., Neyman, J. (Eds.), Proc. 5th Berkley Symp. on Mathematical Statistics and Probability, vol. I. University of California Press. xvii pp. 666.

Meila, M., Heckerman, D., 1998. An experimental comparison of several clustering methods, Microsoft Research Report MSR-TR-98-06, Redmond, WA.

Mitra, P., Murthy, C.A., Pal, S.K., 2002. Density based multiscale data condensation. IEEE Trans. Pattern Anal. Machine Intell. 24 (6), 734–747.

Neter, J., Wasserman, W., Whitmore, G.A., 1992. Applied Statistics. Allyn and Bacon.

Penã, J.M., Lozano, J.A., Larrañaga, P., 1999. An empirical comparison of four initialization methods for the *K*-means algorithm. Pattern Recognition Lett. 20, 1027–1040.

Roy, N., McCallum, A., 2001. Towards optimal active learning through sampling estimation of error reduction. In: Proc. 18th Internat. Conf. on Machine Learning (ICML-2001).

Ruspini, E.H., 1970. Numerical methods for fuzzy clustering. Inform. Sci. 2, 319–350.

Thiesson, B. Meck, C., Chickering, D., Heckerman, D., 1997. Learning mixtures of Bayesian networks, Microsoft Research Technical Report TR-97-30, Redmond, WA.

UCI Repository (http://www.sgi.com/tech/mlc/db/).

Witten, H.I., Frank, E., 2000. Data Mining Practical Machine Learning Tools and Techniques with Java Implementation. Morgan Kaufmann Publishers, San Francisco, CA.

Yi-tzuu, C., 1978. Interactive Pattern Recognition. Marcel Dekker Inc., New York and Basel.