Cluster Feature-Based Incremental Clustering Approach (CFICA) For Numerical Data

A.M.Sowjanya[†] and M.Shashi[††],

Department of Computer Science. & Systems Engineering , College of Engineering ,
Andhra University, Visakhapatnam.

Summary
Data clustering is a highly valuable field of computational statistics and data mining. A major difficulty in the design of modern data clustering algorithms is that, in majority of applications, new data sets are dynamically appended into an existing massive database and it is not viable to perform data clustering from scrape every time new data instances get added up in the database. The development of clustering algorithms which handle the incrementally updated data points, has received increasing interest among the researchers. This paper presents a more efficient cluster feature-based incremental clustering approach (CFICA) for numerical data sets. Initial clustering is performed on the static database with the help of k-means clustering algorithm. Then, by making use of cluster feature computed from the initial clusters, the incrementally updated data points are clustered. Subsequently, the closest pair of clusters is merged to obtain better cluster accuracy. Finally, the proposed approach has been validated with the help of real datasets presented in the UCI machine learning repository. The experimental results demonstrated that clustering accuracy of the proposed incremental clustering approach is improved significantly.

Key words:
Data mining, Clustering, Incremental clustering, k-means algorithm, Cluster Feature, Mean, Farthest neighbor points, Clustering accuracy (CA*)*

1. Introduction

In view of the fast growth of World Wide Web, Internet users have overwhelming quantities of online information that require automatic data mining techniques. Manual analysis of data is a vague process in which quick information retrieval by the user is impossible [1]. Clustering is the unsupervised classification of patterns into groups. It groups a set of objects into different subsets such that objects of the same cluster are highly equivalent to one another. The various applications of clustering include image segmentation, information retrieval, web pages grouping, market segmentation, and scientific and engineering analysis [2].

Data clustering is a primary data mining method and an efficient technique for data analysis. Continuous dumping of each and every raw data set into an existing massive database requires the design of new clustering algorithms. A solution to handle this problem is to integrate a clustering algorithm that functions incrementally [3]. Incremental clustering algorithms permit a single or a few passes over the whole dataset to put the updated item into the cluster. With respect to the size of the set of objects, algorithms and number of attributes, incremental clustering algorithms are of scalable nature [4]. The model of incremental algorithms for data clustering is necessiated by realistic applications where the demand sequence is not known in advance and the algorithm should keep a constantly fine clustering using a restricted set of operations resulting in a solution of hierarchical structure [5].

In this paper, we report a more efficient approach, Cluster Feature-based Incremental Clustering (CFICA). The static database is given to the k-means clustering algorithm for initial clustering. The cluster obtained is used to compute the cluster feature, which consists of mean and p-farthest neighbor points of the cluster. The computed cluster feature is used for clustering the incremental database, wherein the devised distance measure is used to find the appropriate cluster for each incoming data point. At the same time, a new cluster is formed if the computed distance measure is above the predefined threshold level. Subsequently, the cluster feature is updated each time after finding the relevant cluster for every incoming data point. The closest cluster pair for a set of processed data points, is merged using the mean of the cluster. This procedure is repeated for each data point available in the incremental database and finally, the resultant cluster is obtained.

2. CFICA algorithm

Most of the clustering algorithms presented in the literature are used for clustering the static database. Now-a-days, research community has shifted focus to the incremental databases for real-world applications and the necessity of handling dynamically updated data points. Motivated by this research interest, we have developed an incremental clustering approach for numerical data sets. Initial clustering and handling of incremental data points

Manuscript received September 5, 2010
Manuscript revised September 20, 2010

are two important steps of the incremental clustering approaches. We made use of K-means clustering algorithm (Conventional clustering algorithm) for initial clustering and then, the proposed approach, for clustering the incremental data points has been applied. Let S_D be the original data base (static database) and ΔS_D be the incremental database. Our ultimate aim is to cluster the database $S_D + \Delta S_D$, in which the data points from the incremental database ΔS_D are updated over time. The procedure used for the proposed incremental clustering approach is given in figure 1.

Input: Static Database S_D, Incremental Database ΔS_D, Merging Threshold M_T, Threshold

N_T, k, t.

Output: The resultant cluster, C_R

Parameters: Δy : incoming data point

CF : Cluster feature

$D_{\Delta y}^{(i)}$: Distance Measure

m_i : Mean value

E_D : Euclidean Distance

$q_i^{(j)}$: p-farthest neighbor points

Q_i : farthest neighbor point

Method:

1. Call k-mean (S_D, k)
2. For i from 1 to k

   a) Compute CF = $\{m_i, q_i^{(j)}\}$
3. For each point Δy in ΔS_D

   b) For i from 1 to k

      i) Compute $\{m_i, Q_i\}$

      ii) Calculate distance, $D_{\Delta y}^{(i)}$

   c) Add Δy to i^{th} cluster, if min $\left(D_{\Delta y}^{i}\right) < N_T$

   d) Forming a new cluster, if min $\left(D_{\Delta y}^{i}\right) \geq N_T$

   e) Update CF = $\{m_I, q_I^{(j)}\}$

   f) After processing 't' data points in ΔS_D

      i) Compute E_D in between mean points

      ii) Merge the closet cluster pair; if $E_D < M_T$

      - Update the mean of merged cluster
      - Go to step 3.f. (i)
4. The resultant cluster, C_R

Subroutine: k-mean $\left(S_D^*, k^*\right)$

1. Choose initial means m_1, m_2, ……., m_{k^*} from S_D^*
2. Until there are no change in the mean

   a) Use the estimated mean to cluster the data points

   b) For i from 1 to k*

      i. Replace mi with the mean of data points for cluster I

   c) Go to step (2, a)
3. The set of k – cluster

Fig. 1 Proposed Incremental Clustering Approach

2.1 Initial clustering with static database

At first, the clustering is performed on the static database (S_D), where the data points do not change over time. Here, for clustering the static database (S_D), we have used the k-means clustering algorithm so that, k number of clusters are obtained. K-means algorithm is one of the most popular data clustering methods because of its simplicity and computational efficiency.

2.1.1 K-means clustering algorithm

K-means clustering is a well-known partitional algorithm that aims to partition n data points into k clusters, in which each data point belongs to the cluster with the nearest mean Assume the static data base, S_D comprising of *n* data points $y_1, y_2, \cdots, y_n$ so that every data point is in $\mathbf{R}$, the problem of identifying the minimum variance clustering of the dataset into k clusters is none other than finding k means $\{m_i\}$ $(i = 1,2,\cdots,k)$ in $\mathbf{R}$ whereas

$$\frac{1}{n}\sum_{j=1}^{n}\left[\min_{i} d^{2}(y_j, m_i)\right]$$

is minimized, where $d(y_j, m_i)$ indicates the Euclidean distance between y_j and m_i . The points $\{m_i\}\ (i = 1,2,\cdots,k)$ are termed as cluster centroids. The problem in the aforementioned equation is to obtain *k* cluster centroids, in which the average squared Euclidean distance (mean squared error, MSE) between a data point and its nearest cluster centroid is minimized [6].

Steps:

1) Initialize k means, one for each cluster.
2) Compute the distance $E_D(y_j, m_i)$ (described in definition 1) of each k centroid with data points y_j in S_D .
3) Assign data point y_j to cluster C_i whose distance is less.
4) Update the k centroids based on the memberships of new clusters.
5) Repeat Step 2 to step 4, until there is no movement of the data points between the clusters.

Definition 1: (Distance Measure)

For calculating the distance between the centroid and the data points, we make use of the Euclidean distance [7]. The Euclidean distance (E_D) between points y and m is the length of the line segment $\overline{ym}$. In Cartesian coordinates, if $y = (y_1, y_2, \cdots, y_l)$ and $m = (m_1, m_2, \cdots, m_l)$ are two points in Euclidean l -space, afterwards the distance from y to m is formulated by:

$$E_D(y,m) = \sqrt{(y_1 - m_1)^2 + (y_2 - m_2)^2 + \cdots + (y_l - m_l)^2} = \sqrt{\sum_{i=1}^{l}(y_i - m_i)^2}$$

2.2 Clustering of incremental database

We obtain k number of clusters, which are represented in a set, $c = \{c_1, c_2, \cdots, c_k\}; 1 \le i \le k$ from the k-means algorithm using the static database S_D . After that, using the proposed approach, we cluster the incremental database ΔS_D consisting of r data points $\Delta y_1, \Delta y_2, \cdots, \Delta y_r$; $1 \le j \le r$. The motivation behind this approach is that, in incremental database, data points Δy_j are added over time. These changes should be reflected in the resultant cluster C_R without extensively affecting the current clusters, c_i . After the database is updated, there are three possibilities for clustering the updated points as shown in the figures 2 to 4.Assume A and B are the initial clusters for an instance and D represents the new incoming data points.

Case 1: Adding with the existing cluster .

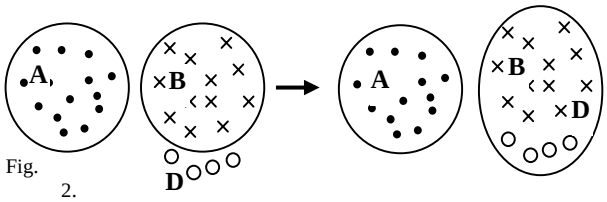

Fig. 2. Representation of adding to the existing cluster

Case 2: Formation of a new cluster.

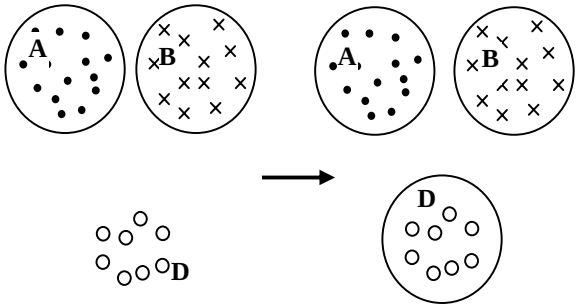

Fig. 3. Representation of formation of a new cluster

Case 3: Possibility to merge the existing clusters when updated points are in between the existing two clusters.

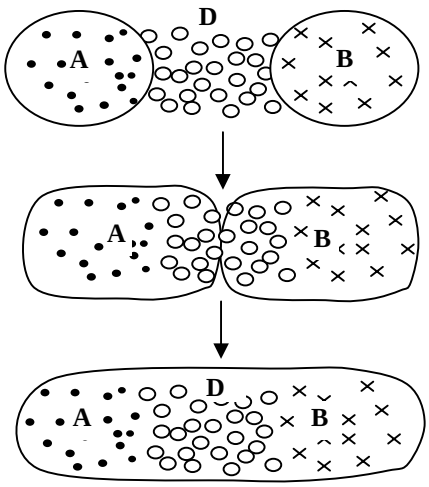

Fig. 4. Representation of merging the closest cluster pair

The proposed approach is designed by making use of the above three possibilities and the steps of the approach is given in this sub-section.

2.2.1 Computation of Cluster Feature (CF)

The cluster feature (CF) is computed for every cluster c_i which is obtained from the k-means algorithm. In the proposed approach, mean and p-farthest neighbor points are used to represent the CF, which is denoted as, $CF_i = \{ m_i , q_i^{(j)} \}; i \le j \le p$, where m_i is the mean of the cluster c_i and $q_i^{(j)} \in \mathrm{S_D}$ gives the p-farthest neighbor points of cluster c_i. The p-farthest neighbor points of the cluster c_i are calculated as follows: The Euclidean distance value E_D is calculated in between the data points within cluster c_i and the mean of corresponding cluster m_i. Then, the data points are arranged in descending order with the help of the measured Euclidean distance. Subsequently, the top p-farthest neighbor points for every cluster are chosen from the sorting list and these points ($q_i^{(j)}$; $i \le j \le p$) are known as p-farthest neighbor points of the cluster c_i with respect to the mean value m_i.

2.2.2 Finding the appropriate cluster

In this step, we initiate the clustering process for dynamically updating data points available in the incremental database, ΔS_D. At first, the farthest neighbor point Q_i which is nearer to the incoming data point Δy is identified for every cluster c_i. We make use of the p-farthest neighbor points for identifying the farthest neighbor point Q_i. In order to find the farthest neighbor point Q_i, we calculate the Euclidean distance E_D described in definition 1, for an incoming data point Δy with each p-farthest neighbor points of every cluster c_i. Then, the p-farthest neighbor points are sorted based on the computed distance measure and thereby, for each cluster, we take one point, known as farthest neighbor point Q_i having minimum distance. After that, we find the distance $D_{\Delta y}^{(i)}$ of every cluster c_i with the incoming data point Δy. The incoming data point Δy is assigned to the current cluster having minimum distance only if the calculated distance is less than the predefined threshold level, N_T. Otherwise, the incoming data point Δy is separately formed as a new cluster so that, the number of cluster is incremented with one.

Distance measure: Distance strategy is an important for effectively identifying the appropriate cluster of an incoming data point Δy. In the proposed approach, we have used different distance strategy that makes use of three points such as mean m, farthest neighbor point Q and the incoming data point Δy. For each cluster c_i, the Euclidean distance E_D is calculated for the following two set of points: mean and incoming point $(m_i, \Delta y)$, farthest neighbor point and incoming point $(Q_i, \Delta y)$, mean and farthest neighbor point (m_i, Q_i). When new data point Δy is updated in the static database S_D, the distance $D_{\Delta y}^{(i)}$ is calculated using the following equation. The cluster quality of the proposed approach is improved significantly if these three measures are incorporated into the distance value for identifying the appropriate cluster.

$$D_{\Delta y}^{(i)} = E_D(m_i, \Delta y) + \left(E_D(Q_i, \Delta y) \times E_D(m_i, Q_i) \right)$$

where, $E_D(m_i, \Delta y)$ → Euclidean distance between the points m_i and Δy; $E_D(Q_i, \Delta y)$ → Euclidean distance between the points Q_i and Δy; $E_D(m_i, Q_i)$ → Euclidean distance between the points m_i and Q_i

2.2.3 Updating of Cluster Feature

After adding the incoming data point Δy to the cluster, updating of CF is important for further processing. The cluster feature (CF_I) of the incremented cluster C_I (the cluster c_i with incoming data point Δy) is updated by : (1) taking the mean of the incremented cluster C_I (2) finding the p-farthest neighbor point of the incremented cluster C_I. For updating of CF, we first calculate the mean of the incremented cluster (m_I) and update the first

component of the cluster feature CF_I with m_I. Then, Euclidean distance E_D is calculated for every data points in the incremented cluster C_I with the updated mean m_I .The data points are sorted based on the computed distance measure and new set of top p -farthest neighbor points are chosen from the sorted list. Such a way, the new cluster feature is formed for the incremented cluster C_I, which is represented as, $CF_I = \left\{ m_I , q_I^{(j)} \right\}$: $i \le j \le p$, This process of updating is applicable only to the incremented cluster C_I which is updated recently. Finding the appropriate cluster (sub-section 2.2.2) and updating of cluster feature (sub-section 3.2.3) are iteratively performed for 't' number of data points in the incremental database ΔS_D.

2.2.4 Merging of closest cluster pair

Once the 't' number of data points in the incremental database ΔS_D are processed with the proposed approach, there is an urgent requirement to merge the closest clusters. To avoid the formation of a non-optimal clustering structure which may occur during the incremental clustering process, the merging strategy is used in the proposed approach that guides the incremental clustering process with high quality and at the same time, the proposed approach generates the reasonable number of clusters.

For merging the closest cluster pair, we make use of the CF employed in the previous steps. The mean of every cluster identified after finishing the 't' number of iteration is used for merging the closest cluster pairs. If "t" data points are processed with the proposed approach, then we perform the merging strategy where, the closest cluster pair is identified by making use of the mean and the Euclidean distance. The procedure used for merging process is described as: (i) Calculate the Euclidean distance E_D in between the mean of the cluster with other cluster (ii) Select the cluster pair having minimum distance (iii) Merge the chosen cluster pair, if the distance of the chosen cluster pair is less than the merging threshold level, M_T (iv) Recalculate the mean of the merged cluster (v) Repeat step (i) to (iv) until no cluster pair is merged. Finally, the resultant cluster C_R is obtained from the merging process after processing all data points in the incremental database.

3. Results

We have implemented the proposed incremental clustering approach using two static datasets Iris dataset [8] and wine dataset [9] from the UCI machine repository. *Iris dataset* comprises 3 classes of 150 instances each, in which each class defines to a type of iris plant. The attributes are sepal length in cm, sepal width in cm, petal length in cm, petal width in cm and class label. *Wine dataset* comprises 3 classes of 178 instances. These data are the outcome of a chemical analysis of wines grown in similar regions in Italy but resulting from three diverse cultivators. The analysis resolves the quantities of 13 constituents identified in each of the three types of wines. The attributes are Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue, Proline and class label.

To convert these static problems into dynamic problems, the data points in each database are divided into 2 groups. each group consisting of identical number of data points from each class. Initially, we give the first group of data points to the k-means algorithm for initial clustering. It provides the k- number of clusters. Then, we compute the cluster feature for initial cluster and by making use of the cluster feature; we fed the second group of data points to the proposed approach incrementally. For each data point from the second group, we compute the designed distance measure and assign the data points to the corresponding cluster if the minimum distance of the cluster is less than the minimum threshold (N_T =10). Otherwise, it forms as a separate cluster. Subsequently, the cluster feature is updated for each data points based on the proposed approach. Once we process the 't' (t =20) set of data points, the merging process is done in accordance with the merging threshold (M_T = 4). Finally, we obtain the set of resultant cluster from the merging process.

The performance of the proposed approach is evaluated on Iris dataset and wine dataset using Clustering Accuracy (CA). We have used the clustering accuracy described in [37, 38] for evaluating the performance of the proposed approach. The evaluation metric used in the proposed approach is given below,

$$\text{Clustering Accuracy, } CA = \frac{1}{N}\sum_{i=1}^{T} X_i$$

$$\text{Clustering Error, } \mathbf{CE = 1 - CA}$$

where, N → Number of data points in the dataset;

T → Number of resultant cluster ;

X_i → Number of data points occurring in both

cluster i and its corresponding class. T

The experimental results of the proposed approach are shown in figure 5 and figure 6. We calculate the clustering accuracy of the resultant clusters by changing the k-value (order of initial clustering) and at the same time, clustering error is also calculated

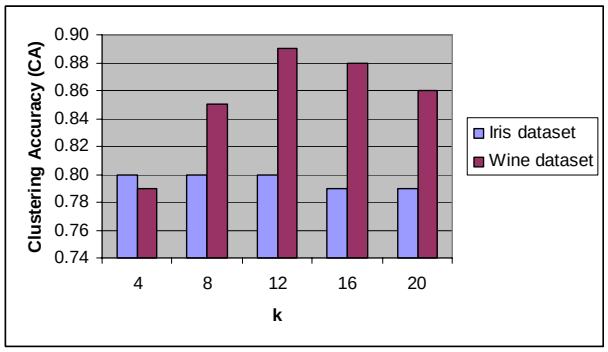

Fig. 5. Clustering Accuracy vs. input order of initial clustering (k)

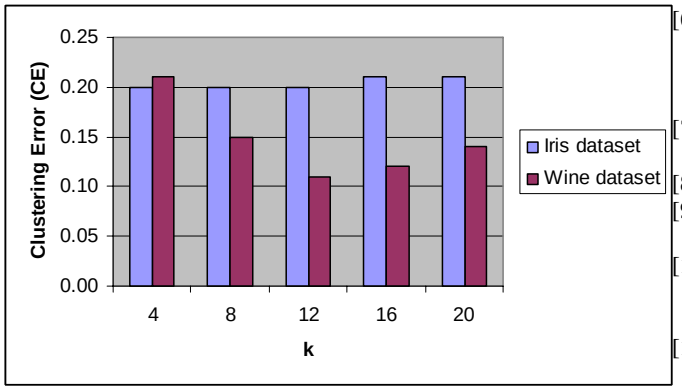

Fig. 6. Clustering Error vs. input order of initial clustering (k)

4. Conclusion

In this paper, we have developed a more efficient approach for clustering incremental database using cluster feature. The proposed approach has two modules namely, 1) initial clustering 2) incremental clustering. First, initial clusters have been obtained by using the conventional partitional clustering algorithm - k-means algorithm then, cluster features have been obtained using the initial cluster and these cluster features have been used for clustering the incremental database in accordance with the devised distance measure. Finally, by making use of the mean value, closest pair of clusters has been merged after processing the set of data points. For experimentation, we have employed the real datasets obtained from the UCI machine learning repository. The experimental results indicated that the proposed incremental clustering approach is efficient in terms of clustering accuracy.

References

[1] Seokkyung Chung and Dennis McLeod, "Dynamic Pattern Mining: An Incremental Data Clustering Approach", Journal on Data Semantics, Vol. 2, pp. 85-112, 2005.

[2] Pham, D.T. and Afify, A.A. “Clustering techniques and their applications in engineering”, Proceedings- Institution of Mechanical Engineers Part C Journal of Mechanical Engineering Science, Vol: 221; No: 11, pp: 1445-1460, 2007.

[3] Chien-Yu Chen, Shien-Ching Hwang, and Yen-Jen Oyang, "An Incremental Hierarchical Data Clustering Algorithm Based on Gravity Theory", Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, Pages: 237 - 250, 2002.

[4] Dan Simovici, Namita Singla, "Metric Incremental Clustering of Nominal Data", ICDM, pp: 523-526, 2004.

[5] Dimitris Fotakis, "Incremental algorithms for Facility Location and k-Median", Theoretical Computer Science, Vol: 361, No: 2-3, pp: 275-313, 2006.

[6] FAHIM A.M., SALEM A.M., Torkey F.A., Ramadan M.A., "An efficient enhanced k-means clustering algorithm", Journal of Zhejiang University science, vol. 7, no.10, pp.1626-1633, 2006.

[7] Euclidean distance from “http://en.wikipedia.org/wiki/ Euclidean_distance”

[8] Iris dataset from “http://archive.ics.uci.edu/ml/datasets/Iris”

[9] Wine dataset from “http://archive.ics.uci.edu/ml/datasets/ Wine”

[10] Zengyou He, Xiaofei Xu, Shengchun Deng, “Clustering mixed numeric and categorical data: A cluster ensemble approach”, abs/cs/0509011.

[11] Z. Huang, “Extensions to the k-means algorithm for clustering large data sets with categorical values”, Data Mining and Knowledge Discovery, vol. 2, no.3, pp. 283-304, September 1998.

[12] Chung-Chian Hsu and Yan-Ping Huang: "Incremental clustering of mixed data based on distance hierarchy", Expert Systems with Applications, Vol: 35, No: 3, pp: 1177-1185, 2008.

[13] Euclidean distance from “http://en.wikipedia.org/wiki/ Euclidean_distance”

[14] Gavin Shaw & Yue Xu, "Enhancing an Incremental Clustering Algorithm for Web Page Collections", Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, Vol: 3, pp: 81-84, 2009.

[15] Jiawei Han, Micheline Kamber, “Data Mining Concepts and Techniques”, Harcourt India Private Limited, 2001.

A.M.Sowjanya received her M.Tech. Degree in Computer Science and technology from Andhra University. She is presently working as an Assistant Professor in the department of Computer Science and Systems Engineering, College of Engineering (Autonomous), Andhra University, Visakhapatnam, Andhra Pradesh, India. She is pursuing her Ph.D from Andhra University. Her areas of interest include Data Mining and Database management systems.

M.Shashi received her B.E. Degree in Electrical and Electronics and M.E. Degree in Computer Engineering with distinction from Andhra University. She received Ph.D in 1994 from Andhra University and got the best Ph.D thesis award. She is working as a professor and HOD of Computer Science and Systems Engineering at Andhra University, Andhra Pradesh, India. She received AICTE career award as young teacher in 1996. She is a co-author of the Indian Edition of text book on "Data Structures and Program Design in C" from Pearson Education Ltd. She published technical papers in National and International Journals. Her research interests include Data Mining, Artificial intelligence, Pattern Recognition and Machine Learning. She is a life member of ISTE, CSI and a fellow member of Institute of Engineers (India).