Australian Journal of Basic and Applied Sciences, 7(7): 54-60, 2013
ISSN 1991-8178

A Multi-Method Incremental Clustering for Dynamic Datasets

[1]Anbukkarasy. G and [2]Sairam. N

[1&2]School of Computing, SASTRA University, Thanjavur, Tamilnadu, India

Abstract: Clustering is considered as a major factor in data mining, because of its features like simplification, pattern detection, data concept construction and also of its unsupervised learning process. Current tools and techniques used in data mining are not well enough to handle the knowledge discovery from large and dynamic databases. To proceed with knowledge discovery, perfect clustering of the data has to be done, so that the pattern formation should be accurate enough to predict the results. This paper analyses about the major clustering methods and its algorithms and an attempt has been made to integrate the three different types of clustering methods (Efficiency and robustness from Hard- Flat and Incremental nature from Hierarchical) as an incremental clustering approach for dynamic datasets. By this incremental method, the efficiency can be improved by decreasing its time complexity.

Key words: Knowledge discovery, Clustering methods, Incremental clustering.

INTRODUCTION

Knowledge discovery in databases (KDD) together with data mining have been attracting a huge amount of researchers, industry, scientific and media in real-time applications. The increasing speed of the large and enormous databases in recent life makes it hard for the human being to analyze and extract the knowledgeable information from the data, even though the analysts use the classic statistical (data mining) tools. The eventual reason for this enormous amount of data is to attain the unidentified patterns for predicting the informative knowledge, which helps in decision making process.

This can only be achieved by incorporating the appropriate prior knowledge and interpretation of data, rather than just going for the normal data mining techniques. This interpretation can be done by the tools and techniques in KDD, which is mostly needed for artificial intelligence and machine learning researchers. KDD includes techniques like description and prediction to extract the useful information.

Description provides the explicit information in a readable form and can be easily understandable by the user. Prediction is done based on the description of data, which is used to predict the future knowledge. It is easy to predict the knowledge from a small database. But while considering the large heterogeneous dataset, clustering is needed to analyze the data. We made an attempt to go for multi-method approach [20]; to combine two different clustering algorithms together to improve the clustering technique and therefore the knowledge discovery also becomes easier.

1. 1. Clustering:

Clustering can be used as an unsupervised learning technique (learning from raw data); it deals with forming a similar grouping of data as a cluster. Cluster in a database can be defined as “A grouping of related items stored together for efficiency of access”. This clustering can also be done in a more effective manner, when it deals with semi-supervised or supervised learning.

Clustering algorithms can be broadly classified into four categories:

1. Flat clustering – This technique clusters the data without any explicit connection structure with other clusters. That is, there will be no relation among the clusters. Even though the efficiency is considered as the major factor here, predefined input, unstructured clustering and its non-deterministic nature are its limitations.
2. Hierarchical clustering – When efficiency is not considered as a primary concern, we may go for the hierarchical clustering. This overcomes the drawbacks in flat clustering like its unstructured cluster and it is deterministic in nature. The incremental concept of this technique reduces the time complexity in dynamic dataset. It doesn’t work well for clustering the overlapping data objects.
3. Hard clustering – This technique clusters the data when the specified data exactly belongs to one cluster. Even though random partitioning is carried out, the resulting clusters will be more efficient and robust. Efficient clustering of overlapping points can be done and it improves the performance.
4. Soft clustering – This is just contrary to hard clustering where the data may belong to more than one cluster.

Clusters can be validated by its quality and dynamic nature (scalability) of the databases.

Corresponding Author: Anbukkarasy. G., School of Computing, SASTRA University, Thanjavur, Tamilnadu, India.

- Quality - Quality in clustering depends on the internal and external measures. Intra-Cluster tightness and Inter-Cluster separation falls in the internal quality measures, whereas the calculating the data points which wrongly falls in the cluster comes under external measures.
- Scalability - Another factor which affects the cluster is the need for the periodic update of the data in the maintenance phase, which can be dealt with the incremental clustering concept.

The main contribution of this paper includes

- About the Hard-Flat (Section 3.3) and hierarchical (Section 3.4) clustering methods and its algorithms.
- In Section 3.5, we made an attempt to combine the Hard-Flat clustering along with the Hierarchical incremental clustering (Kyuseok *et al*., 2000) method.
- Experimental results of the incremental clustering method will be explained in section 4, by using k-means algorithm with the incremental algorithm.

Related Work:

Danyang Cao and Bingru Yang, 2010 suggested a modified k-Medoids algorithm, to overcome the disadvantages like time complexity, scalability on large datasets called CFk-medoids clustering algorithm. This algorithm works based on the BIRCH algorithm's clustering features and also in the outlier's problem. Clustering Feature is 3-D vector which briefly provides information related to the clusters. This algorithm improves the clustering quality and also the outliers are removed to increase the execution by taking out the outliers in the Clustering Feature tree. Weiguo Sheng and Xiaohui Liu, 2004 suggested to improve the k-medoids algorithm by combining the k-medoids with genetic algorithm, which helps to improve performance and efficient clustering.

Rehab F.Abdel-Kader, 2010 attempts to overcome the disadvantages of k-Means like its sensitiveness to the collection of preliminary partitions and its convergence to the local optimal search by proposing the GAI-PSO k-means clustering algorithm (Genetically-Improved Particle Swarm optimization). They combined the concept of global optimum exploration of the evolutionary algorithms and the quickest convergence of k-means. Velocity factor and its updating rules for the whole dataset are combined in PSO with its initial approximation factors to improve the efficiency of k-means.

Liang sun *et al*., 2010 proposed the k-means based hybrid clustering algorithm. Hybrid clustering here includes the k-Means with the support vector clustering (SVC) algorithm. SVC is not well suited to organize the more overlapping type of cluster structures. K-Means lacks in clustering, when the data is of complex structure or in unknown shape. When combining these two algorithms, the weakness of one algorithm can be eliminated by the other and thus improves the quality of the cluster and therefore this method is more effective.

XING Xiao-shuai *et al*., 2010 proposed the Immune programming algorithm, which defeats the local searching property of k-means with its global searching optimal property. This feature is extracted from the evolutionary programming concept. And in this algorithm by using the prior knowledge, analyzing the non-related data can be avoided and thus improves the speed of the calculation and it is more robust

Ming-Yi shih *et al*., 2010 attempt to combine and cluster two different features (Numerical and Categorical) of data. The categorical attributes are processed to construct the relationship among them based on the co-occurrence. And then based on the relationship, the categorical attributes are converted into numeric attributes and clustering is done. After the TMCM algorithm, HAC (Hierarchical Agglomerative Clustering) and k-means algorithm are integrated and used in this paper for clustering.

David Littau and Daniel Boley, 2009 suggested a scalable method to cluster datasets that are difficult to adapt in memory. PMPDDP (Piecemeal Principal Direction Divisive Partitioning) algorithm is used, in which the original data is broken up into sections to fit in memory. The sections are clustered using PDDP and the distributed product representations are collected and final clustering is done. PMPDDP is more flexible for memory allocation, which increases the accuracy of the clustering. But it does suffer from an increased time cost due to the intermediate clusters computation.

Sudipto Guha *et al*., 2000 deals with the multi-feature data analysis by using the ROCK (Robust Clustering using links) algorithm, which utilizes the connections and not the distance measures when clusters are formed. This presents a good quality cluster and also exhibits good scalability properties. If the number of connections between the pair of data increases, it leads to the increase in the likelihood for constituting in the equivalent cluster. Thus by using the links, the clustering process was injected by the global knowledge.

Fazli Can, 1993 proposed the concept of incremental clustering for dynamic information processing by the C^2ICM (Cover- co-efficient based Incremental Clustering for Maintenance) algorithm. The author suggests rather than just forming the clusters, cluster maintenance is also needed for dynamic information. Here the cluster-splitting approach is used to insert/update a new data into a cluster by reducing the time complexity.

Mu-Chun Su and Chien-Hsing Chou, 2001 proposed a enhanced version of k-Means, where the distance metric is calculated with symmetric value of the cluster. That is, after the initial centroid calculation is done by

Euclidean distance metric, fine tuning is done based on the cluster point symmetry. The flexibility achieved by the SBKM (Symmetry-based version of k-Means) algorithm increases the computational complexity.

Dwi H.Widyantoro *et al.*, 2008 proposed a novel Incremental Hierarchical clustering (IHC) algorithm, which mainly deals with the homogeneity and monotonicity properties. This method considers the data as a tree-structure and it focuses on the inefficiency of clustering in dynamic environment.

Gabriela Serban and Alina Campan, 2005 suggested a new Hierarchical Core-Based Incremental Clustering (HCBIC) algorithm. When a new data enters into the database, repartitioning of the cluster objects are carried out. The decrease in the number of iterations reduces the time complexity

Sophoin Khy *et al.*, 2008 projected a Novelty-based incremental clustering, where the novelty the authors mentioned in the paper represents a similarity function and a variant of k-Means. They suggested a forgetting factor for incremental clustering, which is done based on the F^2ICM (Forgetting Factor-Based Incremental Clustering Method).

M.Srinivas and C.Krishna mohan, 2010 proposed a Leaders complete linkage (LCL) clustering algorithm for hierarchical and incremental clustering methods and they analysed the results along with the Agglomerative Single-link clustering algorithm (ASLCA). By using the LCL, the inter-distance and intra-distance measures are calculated in an efficient way and it helps to improve the quality of clusters.

Proposed Method & Algorithm:

3. 1. Hard-Soft clustering:

In hard clustering, disjoint partition of the data is done; so that each data point exactly belong to only one of the cluster. Soft clustering can be defined as each data point has a certain probability of belonging to each of the partitions. Hard clustering can be called as soft clustering when the probabilities are either 0 or 1, whereas they can take any non-negative values in soft computing.

3. 2. Flat clustering:

Flat clustering creates a group of clusters with only an implicit structural connection, that would relate clusters to each other and strictly it should not have explicit relationship among the clusters. This method is similar to hard clustering, but it is allotted to a cluster having a single type of data attribute alone.

3. 3. Hard-Flat Clustering:

Definition:

Given a set of data D = {d_1, d_2, …, d_k}, user-defined number of clusters (k), where the cluster is denoted by (C) and the function (Ƒ) to evaluate the quality of cluster, where

C_i =D → {1,2, .. , k}

Where the minimized metric function is considered and none of the clusters should be empty. The Hard-Flat clustering (21] uses the metric function, which is calculated based on the distance between objects.

Procedure:

i. Choose a flat dataset (strictly a single data attribute).
ii. Randomly choose the centroid from the dataset.
iii. Calculate the function (distance measure) of the data with each of the centroid.
iv. Form the cluster with the minimum values.
v. Repeat step (ii), (iii) and (iv) till terminating condition occurs.

Hard-Flat Algorithm:

```
Initialize $\{\mu_h\}_{h=1}^{k}$
While (Converged)
Assign each data point x, to the nearest cluster $X_h$, such that
h= argmin$d_\psi$(x, $\mu_s$ )
Re-estimate the representatives.
$\mu_h = \frac{\sum_{x \epsilon X_k} x}{n_{x_k}}$
```

Complexity:

The time complexity for Hard-Flat clustering is O(nkmi), where

- n → Number of data objects
- m → Distance measure calculation
- k → Number of centroids
- i →Number of iterations

For this Hard- Flat clustering, k-Means is considered as the best algorithm because of its simplicity and efficiency. It suits for Hard clustering because the data objects will strictly fall within a single cluster and its nature of adapting only a single attribute makes it suits for flat clustering.

3.4. Hierarchical clustering:

Definition of Hierarchical clustering:

Hierarchical clustering is considered as a greedy algorithm, where the concept of merging (Agglomerative) and splitting (Divisive) is used to cluster the objects. The result of clusters will be like a hierarchy of clusters; a Tree-like structure (Dendogram). This doesn't need the cluster size or the initial centroid for clusters. All this algorithm needs is the measure of a similarity between the data objects. The similarity measure of inter-cluster data objects can be calculated by single-linkage, complete-linkage and group-average clustering methods.

Procedure:

i. Assign each data object as a single cluster.
ii. Similarity measure (distance) is calculated for each of the clusters with its neighbours.
iii. Compute distance of the new cluster with each of its cluster.
iv. Repeat steps (ii) and (iii) till the similarity measure gets maximum.

EfficientHAC algorithm:

Input: Dataset D → (d_1, d_2, …., d_T)

```
For t ← 1 to T do
    For m ← 1 to T do
  //si ← similarity measure
        Cluster[t][m].si ← $d_t.d_m$
  //ind ← index
        Cluster[t][m].ind ← Index
    Index[t] ← 1
    Pri[t] ← Priority queue for Cluster[t] based on si
// No need for self-similarities
    Pri[t].Delete(Cluster[t][t])
E ← []
For P ← 1 to T-1 do
    $P_1$ ← $\text{argmin}_{\{p.Index[T]=1\}}$ Pri[P].Max().si
    $P_2$ ← Pri[$P_1$].Max().ind
    E.Append(<$P_1$, $P_2$>)
    Index[$P_2$] ← 0
    Pri[$P_1$] ← []
    For each i with Index[i] =1 ^ i ≠ $P_1$ do
        Pri[i].Delete(C[i][$P_1$])
        Pri[i].Delete(C[i][$P_2$])
        Cluster[i][$P_1$].si ← si (Index, $P_1$, $P_2$)
        Pri[i].Insert (Cluster[i][P1])
        Cluster[$P_1$][i].si ← si (Index, $P_1$, $P_2$)
        Pri[$P_1$].Insert ( Cluster[$P_1$][i])
Return E
```

Similarity measures[13]:

Single-Linkage (Minimum/ Connectedness)	Min $_{di€G,\ j€H}$ d (i,j)
Complete-Linkage (Maximum/ Farthest)	Max $_{di€G,\ j€H}$ d (i,j)
Group-Average	$\frac{1}{N_G N_H}\sum_{i€G}\sum_{j€H} d(i,j)$

Complexity: In hierarchical agglomerative, for worst case it is O (n^3) and for best case, it is O (N^2 log N).

Hierarchical clustering can be also defined as incremental clustering because of its reduced time complexity and non-determined cluster size. It is well suited for dynamic real time dataset.

3.5. Incremental Clustering method:

K-Means is well suited for static type of huge data, but it is not suited well enough for dynamic data. In the same way, hierarchical clustering is well-matched for incremental concept, but it is not well-efficient for large database. So, we are proposing an incremental clustering concept which merges the advantage of k-means with the incremental concept.

3.5.1. K-Means algorithm:

K-Means is considered as simple and efficient unattended type of algorithm, even though it lacks in global searching of objects, increase in time complexity in real-time environment (dynamic nature). The distance measure here is calculated based on the Euclidean distance formula. Since this paper focuses about the numeric type of data attributes, one-dimensional formula of Euclidean distance metric is used.

$D(x, y) = |x-y|$

Where D (x, y) is the measure of distance between two points x (centroid of clusters) and y (data)

$ɳ_k \leftarrow \frac{1}{|\omega^k|}\sum_{x\varepsilon\omega_k} x$

Centroid is re-calculated by using the above average condition. When these centroid calculations and re-computation of clusters are made in parallel, the time complexity can be reduced.

Algorithm:

Input: Training set of data (D), Number of Clusters (S)

Output: Prediction of data items for each Cluster (C_s) and Final Centroids ($ɳ_s$).

```
(C_1, C_2, …, C_s) ← SelectRandomCentroid({d_1, d_2, …, d_s}, S)
For s = 1 to S Do
$ɳ_s \leftarrow C_s$
While termination condition is not reached Do
 For s ← 1 to S Do
     $\Psi_s \leftarrow \{\}$
     For n ← 1 to N do
     $k \leftarrow \min_k (|ɳ_k - d_n|)$
// Re-computation of clusters
     $\Psi_s \leftarrow \Psi_s \cup \{ d_n \}$
     For k ← 1 to K Do
// Re-computation of centroids
      $ɳ_s \leftarrow \frac{1}{|\omega^s|}\sum_{x\varepsilon\omega_s} x$
Return { $ɳ_1, ɳ_2, …, ɳ_s$ }
```

Termination condition can be considered based on the any of the user perspectives like:

i. Fixed number of clusters.
ii. Data should be same within the cluster structures for different iterations.
iii. Centroids should be same for different iterations.
iv. Setting a threshold value in its minimum Euclidean distance metric.

3.5.2. Incremental Clustering:

Then we are going to integrate the incremental concept with k-means algorithm, to make it suitable for dynamic datas*et al*so.

Procedure:

1 The initial set of data is clustered by means of k-means.
2 Calculate the distance measure of the new data with each of the centroid.
3 If (Incremental)
4 Insert the data with its minimum distance value to the cluster and dataset separately, so that it won't affect the static clustering done by k-means.
5 If (Decremental)
6 Find the cluster by its minimum distance value
7 Delete the data from the cluster and dataset separately.
8 Else if (Update)
9 Perform deletion and then insertion

Algorithm:

```
Incremental ( )
D ← { d_n } U d
For k ← 1 to K do
        $j \leftarrow \min_j (|\mu_j - d_n|)$
        $\Psi_k \leftarrow \Psi_k \cup \{ d_n \}$
Decremental ( )
D ← { d_n } - d
For k ← 1 to K do
```

```
        $j \leftarrow \min_j (| \mu_j - d_n |)$
        $\Psi_k \leftarrow \Psi_k - \{ d_n \}$
Update ( )
    Decremental ( );
    Incremental ( );
```

By merging these concepts, we can reduce the time complexity and computation complexity of both the clustering methods.

Experimental Analysis:

Let us consider a dataset of having data in its minimal thousands and execute it in two different forms. One is to run the code for Hard-Flat clustering (k-means) alone and the execution times are calculated. And the second option is to run by our incremental clustering concept, where the initial clustering is done by k-means and then incremental method is carried out.

Table 1: Analysis for k-means and incremental method.

Data Increment	Execution time (Milliseconds)		
	k-Means	Incremental	Time savings
0 to 403	435	-	
403 to 500	522	102	420
500 to 653	762	187	575
653 to 859	993	252	741
Total time savings			1736

Table 1. shows the comparison of k-means and incremental method by considering its execution time. The execution time is calculated for the dynamic dataset; incrementing the data in the dataset during execution.

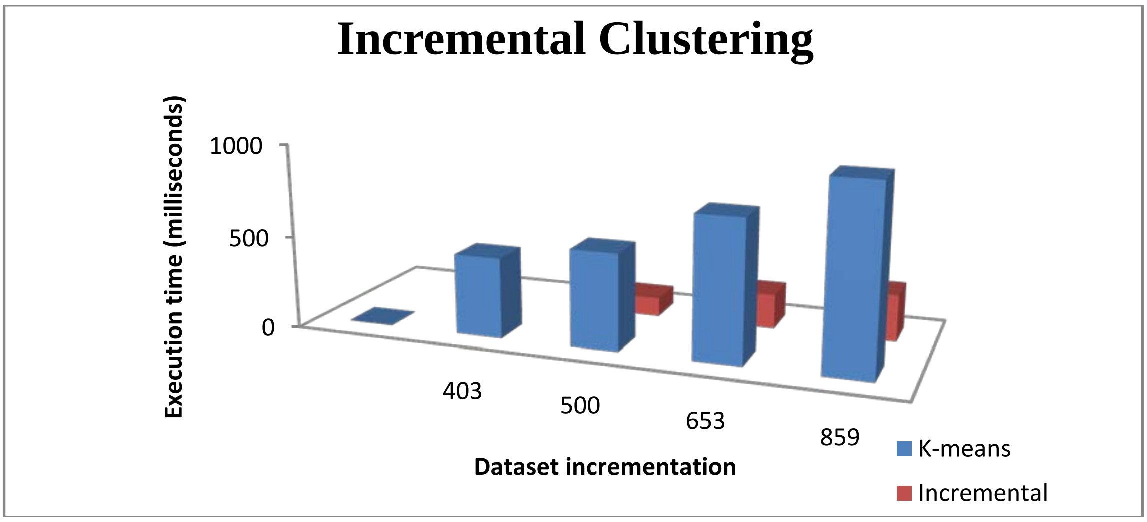

Fig. 1: Analysis of incremental clustering

In Figure 1, the execution time (in milliseconds) is calculated across the data. By using the k-means, initial 403 data are clustered together in 435ms. When the clustering is done for every increment of data till near some thousands of data, the execution time goes around 1000ms. In our incremental method, the same thousands of data can be executed in a short duration by nearly reducing 3/4th of the execution time of k-means. From this experimental result, we can conclude that even for a huge data, the time will be much reduced and therefore the efficiency will be increased to a great extent.

Conclusion:

Clustering is considered as an important method to retrieve the information, while working in a large dataset. A perfect clustering of dataset can have the ability to make the entire prediction in a more easy way. So rather than using a single clustering method, an integration attempt is carried out in this paper. The experimental results shows that the time variation of the normal Hard-Flat clustering with the Incremental method and how well the time complexity is reduced. By this method, the efficiency can also be improved in predicting the knowledge.

REFERENCES

Li zhu, Xing Xiao-shuai, Yang Pei-lin, Yao Jian-bin and Zhang Qing-quan, Oct. 2010 “A Novel Hybrid Clustering Algorithm Incorporating k-means into canonical Immune Programming Algorithm”, IEEE International Conference on Multimedia Technology (ICMT), 1-4,.

Jar-Wen Jheng, Lien-fu Lai and Ming-Yi Shih, 2010 “A Two-Step Method for Clustering Mixed Categorical and Numeric data”, Tamkang Journal of science and Engineering, 13(1): 11-19.

Bruno Stiglic, Milan Zorman, Mitja Lenic, Peter Kokol, Petra Povalej, and Ryuichi Yamamota, 2005 “Improved Knowledge Mining with the Multimethod approach”, Studies in Computational Intelligence (SCI), 6: 305-318.

Kyuseok shims, Rajeev Rastogi and Sudipta guha, 2000 “ROCK: A Robust Clustering Algorithm for Categorical attributes”, Information systems, 25(5): 345-366.

Arindam Banerjee, Inderjit S.Dhillon, Oydeep Ghosh and Srujana Merugu, “Clustering with Bregman Divergences”, Csci 8980: Machine Learning.

Fazli can, April 1993 “Incremental clustering for dynamic information processing”, ACM transactions on information systems, 11(2): 143-164.

Chien-Hsing Chou and Mu-Chun Su, June 2001 “A modified version of the K-Means algorithm with a distance based on cluster symmetry”, IEEE transactions on pattern analysis and machine intelligence, 23(6).

Weiguo Sheng and Xiaohui Liu, June 2004 “A Hybrid algorithm for K-Medoid Clustering of large datasets”, IEEE Congress on Evolutionary Computation (CEC), 1: 77-82.

Liu Lu, Tian Jinlan, Zhang Suqin and Zhu Lin, June 2005 “Improvement and parallelism of k-means clustering algorithm”, Tsinghua Science and Technology, 10(3): 277-281.

Alina Campan and Gabriela S, Erban, Sep. 2005 “A New Core-Based Method for Hierarchical Incremental Clustering”, Seventh International Symposium on Symboli and Numeric Algorithms for Scientific Computing (SYNASC).

Hiroyuki Kitagawa, Sophoin Khy and Yoshiharu Ishikawa, Apr. 2006 “Novelty-based Incremental document clustering for on-line documents”, IEEE proceedings of the 22nd International Conference on Data Engineering Workshops (ICDEW), 40.

David M.Blei, Feb. 2008, “Hierarchical clustering”, Princeton university.

Haibo He and Yuan Cao, June 2008, “Learning from Testing Data: a New view of Incremental Semi-Supervised Learning”, IEEE International Joint Conference on Neural Networks (IJCNN), 2872-2878.

Dwi H.Widyantoro and Thomas R.Ioerger, Dec. 2008, “An incremental approach to building a cluster hierarchy”, IEEE International Conference on Data Mining (ICDM), 705-708.

“Clustering methods”, April 2009 Cambridge University press.

Daniel Boley and David Littau, May 2009 “Clustering very large datasets using a Low Memory Matrix”, preliminary version of paper in Computational Intelligence, 25(2): 114-135.

Bingru Yang and Danyang Cao, Feb. 2010 “An improved k-medoids clustering algorithm”, Computer and Automation Engineering (ICCAE), The 2nd International Conference, 3: 132-135.

Rehab F.Abdel-Kader, Feb. 2010 “Genetically improved PSO algorithm for Efficient Data Clustering”, IEEE Second International Conference on Machine Learning and Computing, 71-75.

Liang Sun, Shinichi Yoshida and Yanchun Liang, June 2010 “A Novel Support Vector and K-Means based Hybrid Clustering Algorithm”, IEEE International Conference on Information and Automation, 126-130.

C.Krishna Mohan and M.Srinivas, July 2010 “Efficient Clustering Approach using Incremental and Hierarchical clustering methods”, IEEE International Joint Conference on Neural Networks (IJCNN), 1-7.

Lishen Yang, Qianqian Guo, Yingying Li and Yongli Liu, Apr. 2012 “Research on Incremental clustering”, IEEE 22nd International Conference on consumer Electronics, Communications and Networks (CECNet), 2803-2806.