ISBN: 979-756-061-6

PERFORMANCE ANALYSIS OF PARTITIONAL AND INCREMENTAL CLUSTERING

Zuriana Abu Bakar, Mustafa Mat Deris, and Arifah Che Alhadi
University College of Science and Technology Malaysia
Faculty of Science and Technology
21030 Mengabang Telipot, Kuala Terengganu, Malaysia
E-mail: {zuriana, mustafa, arifah_hadi}@kustem.edu.my

Abstract

The partitional and incremental clustering are the common models in mining data in large databases. However, some models are better than the others due to the types of data, time complexity, and space requirement. This paper describes the performance of partitional and incremental models based on the number of clusters and threshold values. Experimental studies shows that partitional clustering outperformed when the number of cluster increased, while the incremental clustering outperformed when the threshold value decreased.

Keywords: *Clustering, partitional, incremental, distance.*

1. INTRODUCTION

Data mining, as one of the promising technologies since 1990s, is some to extent a non-traditional data driven method to discover novel, useful, hidden knowledge from massive data sets [2]. Several data mining tasks have been identified and one of them is clustering. Clustering techniques have been applied to a wide variety of research problems such as in biology, marketing, economics and others.

Clustering is similar to classification in that data are grouped. However, unlike classification, the groups are not predefined. Instead, the grouping is accomplished by finding similarities between data according to characteristics found in the actual data. The groups are called clusters [5].

This paper discussed about partitional and incremental clustering from data mining perspective. The main inherent idea is to compare those clustering techniques to determine which clustering technique is better based on the number of cluster and threshold value. There are many types of data in clustering such as interval-scaled variables, binary variables, nominal, ordinal, and ratio variables. However, in our clustering analysis, only numerical data will be considered.

The rest of this paper is organized as follows. Section 2 discuss related work on clustering techniques. The formulas and algorithms for partitional and incremental clustering are presented in Section 3 and extensive performance evaluation is reported in section 4. Section 5 concludes with a summary of those clustering techniques.

2. RELATED WORK

In this section we provide a brief overview of clustering techniques. Different approaches to clustering data can be described with the help of the hierarchy shown in Figure 1[1].

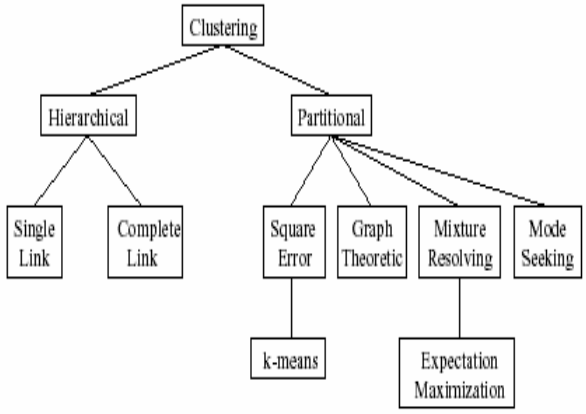

Figure 1. A taxonomy of clustering approaches

Figure 1 illustrates that there is a distintion between hierarchical and partitional approaches. Hierarchical methods produce a nested series of partitions, while partitional methods produce only one.

Hierarchical clustering is often portrayed as the better quality clustering approach, but is limited because of its quadratic time complexity. In contrast, K-means (patitional method) and its variants have a time complexity which is linear in the number of documents, but are thought to produce inferior clusters. Sometimes K-means and agglomerative hierarchical approaches are combined so as to "get the best of both worlds" [8].

Recently, several clustering algorithms for mining in large database have been developed such as Hierarchical Clustering Algorithms, Mixture-Resolving and Mode-Seeking Algorithms Nearest Neighbor Clustering, Fuzzy Clustering etc. This paper focus on the partitional and incremental

ISBN: 979-756-061-6

clustering techniques. There are a number of partitional techniques, but we shall only describe the K-means algorithm which is widely used in data mining.

The K-means partitional clustering algorithm is the simplest and most commonly used algorithm by employing a square-error criterion. It is applicable to fairly large data sets but it is possible to accommodate the entire data the main memory. Besides that, the *k*-means clustering algorithm may take a huge amount of time. Furthermore, K-means finds a local optimum and may actually miss the global optimum [7].

Thus, an incremental clustering algorithm is employed to improve the chances of finding the global optimum and data are stored in the secondary memory and data items are transferred to the main memory one at a time for clustering. Only the cluster representations are stored permanently in the main memory to alleviate space limitations [5].

Therefore, space requirements of the incremental algorithm is very small, necessary only for the centroids of the clusters and this algorithm is non-iterative and therefore their time requirements are also small. But, even if we introduce iterations into the incremental- clustering algorithm, computational complexity and corresponding time requirements do not increase significantly.

3. CLUSTERING ANALYSIS

Cluster analysis is a technique for grouping data and finding structures in data. The most common application of clustering methods is to partition a data set into clusters or classes, where similar data are assigned to the same cluster whereas dissimilar data should belong to different clusters. [6]

An important issue in clustering is how to determine the similarity between two objects, so that clusters can be formed from objects with a high similarity to each other [4].

Commonly, the distances can be based on a single dimension or multiple dimensions. It is up to the researcher to select the right method for his/her specific application. For this clustering analysis, Manhattan distance is being used because the data are single dimension. The Manhattan distance is computed as [4]:

$$distance(x,y) = \sum_i |x_i - y_i|$$

3.1 Partitional Clustering Algorithm

The *k*-means algorithm is one of a group of algorithms called partitioning clustering algorithm**.** The most commonly use partitional clustering strategy is based on square error criterion.

The general objective is to obtain the partition that, for a fixed number of clusters, minimizes the total square errors. Suppose that the given set of *N* samples in an *n*-dimensional space has somehow been partitioned into *K*-clusters $\{C_1, C_2, C_3 \ldots C_K\}$. Each C_K has n_K samples and each sample is in exactly one cluster, so that $\sum n_K = N$, where $k=1 \ldots K$. The mean vector M_k of cluster C_K is defined as the centroid of the cluster or [7]

$$M_K = (1/n_k) \sum_{i=1}^{n_k} x_{ik}$$

Where x_{ik} is the i^{th} sample belonging to cluster C_K. The square-error for cluster C_K is the sum of the squared Euclidean distances between each sample in C_K and its centroid. This error is also called the *within-cluster variation* [7]:

$$e_k^2 = \sum_{i=1}^{n_k} (x_{ik} - M_k)^2$$

The square-error for the entire clustering space containing K cluster is the sum of the within-cluster variations [7]: $E_k^2 = \sum_{k=1}^{K} e_k^2$

The basic steps of the K-means algorithm are:

a. select an initial partition with K clusters containing randomly chosen sample, and compute the centroids of the clusters,
b. generate a new partition by assigning each sample to the closest cluster centre,
c. compute new cluster centre as the centroids of the clusters,
d. repeat steps 2 and 3 until optimum value of the criterion function is found or until the cluster membership stabilizes.

Algorithm 1. *K-means clustering algorithm*

```
Input:
       D = {t1, t2, t3 …, tn}  //set of
elements
       K                       //number of
desired clusters
Output:
       K       //set of cluster
Clustering algorithm:
       Assign each item ti to a cluster
randomly;
       Calculate mean for each cluster;
       Repeat:
              Assign each item ti to the
              cluster which has the
              closet mean;
              Calculate new mean for
each cluster;
              Calculate square error;
       Until
              The minimum total square
errors are reached.
```

ISBN: 979-756-061-6

Algorithm 1 shows the k-means clustering algorithm. Note that the initial values for the means are arbitrarily assigned. These could be assigned randomly or perhaps could use the values from the first *k* input items themselves. The convergence criteria could be based on the squared error, but they need not be [5].

3.2 Incremental Clustering Algorithm

An incremental clustering approach is the way to solve the problems that arise from partitional clustering. Incremental clustering could improve the chances of finding the global optimum. This involves careful selection of the initial clusters and means. Another variation is to allow clusters to be split and merged. The variance within a cluster is examined, and if it is too large, a cluster is split. Similarly, if the distance between two cluster centroids is less than a predefined threshold value, they will be combined. The following are the global steps of the incremental clustering algorithms [5].

a. Assign the first data item to the first cluster.
b. Consider the next data item. Either assign this item to one of the existing cluster or assign it to a new cluster. This assignment is done based on some criterion, e.g., the distance between the new item and the existing cluster centroids. In that case, after every addition of a new item to an existing cluster, recomputed a new value for the centroid.
c. Repeat step 2 till all the data samples are clustered.

Algorithm 2 shows the incremental clustering algorithm. This algorithm is similar to the single link technique called the *nearest neighbor algorithm.*

Algorithm 2. *Incremental clustering algorithm*

```
Input:
        D= { t1, t2, …, tn } // Set of
elements
        A               // Adjacency matrix
showing distance between elements
Output:
        K               // Set of clusters
Nearest neighbor algorithm:

        K1 = { t1 };
        K  = { K1 };
        k   =  1;

for i = 2 to n do
        find the tm in some cluster Km  in
        K such that dis ( ti, tm ) is the
        smallest;
                if dis ( ti, tm ), ≤ t then
                                Km = Km U ti
                Else
                K = k+1;
                Kk = { ti };
```

With this serial algorithm, items are iteratively merged into the existing clusters that are closest. In this algorithm a threshold value, T, is used to determine if items will be added to existing clusters of if a new cluster is created.

4. PERFORMANCE EVALUATION

This section, compared the efficiency of the partitional and incremental clustering. The implementation of both algorithms is using Visual Basic 6.0 and Microsoft Access as its database. Through the performance evaluation, we are going to show that the partitional clustering technique was depends on the number of cluster while as incremental clustering technique depends on the threshold value to get the lower total square error.

This analysis is based on our observation of the air pollution data taken in Kuala Lumpur on the August 2002. A set of air pollution data items consists of five major aspects that can cause the air pollution, i.e. {Carbon Monoxide (CO), Ozone (O_3), Particulate Matter (PM_{10}), Nitrogen Dioxide (NO_2) and Sulfur Dioxide (SO_2)}. The value of each item is with the unit of *part per million (ppm)* except PM_{10} is with the unit of *micro-grams (μgm)*. The data were taken for every one-hour every day. We present the actual data as the average amount of each data item per day. The example of air pollution data is shown in Table 1 below:

Table 1. Air Pollution Data

Date	*CO*	*O_3*	*PM_{10}*	*NO_2*	*SO_2*
1/8/02	2.26	0.010	74	0.005	0.041
2/8/02	2.46	0.120	68	0.004	0.037
…..	…..	…..	…..	…..	…..
30/8/02	2.05	0.012	60	0.006	0.029

In the performance evaluation, both techniques involves computation of centroid where this centroid will be used to cluster the data. In partitional clustering (k-means) the clusters are taken to be defined by their centres meaning that the mean of the coordinates of the elements in the cluster. An element is in the cluster defined by the centre closest to the element. The number of clusters (k) is known. So the space of all possible clustering is the space of k points in the sample space [3] while as in incremental, the value of first data is assume as first centroid.

Table 2. Partitonal Clustering Result

Number of cluster	*Total Square Error*
2	19
3	9
4	5
5	3

ISBN: 979-756-061-6

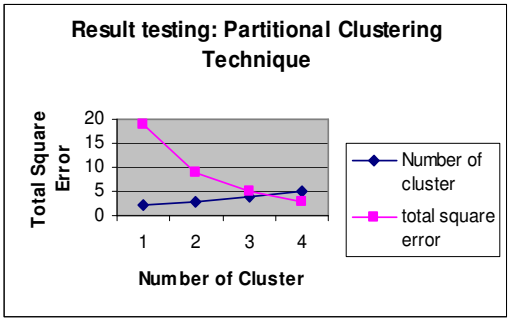

Figure 2. Graph for Partitional Clustering Result

As illustrated in Table 2 and Figure 2, the total square error decreases with the increases in the number of cluster used. This implies that, the lower total square error, the better the clusters would be since the distribution of the data in clusters becomes more compact. In partitional clustering, every data sample is initially assigned to a cluster in some (possibly random) way. Samples are then iteratively transferred from cluster to cluster until some criterion function is minimized. Once the process is complete, the samples will have been partitioned into separate compact clusters.

An Incremental clustering is different with partitional clustering since the data in clusters are fixed. In this technique, the threshold value has to be assigned. This value indicates the distance between the centroid and the data in that particular cluster.

Table 3. Incremental Clustering Result

Threshold	*Total Square Error*
0.2	4
0.5	5
0.7	22
1.2	31

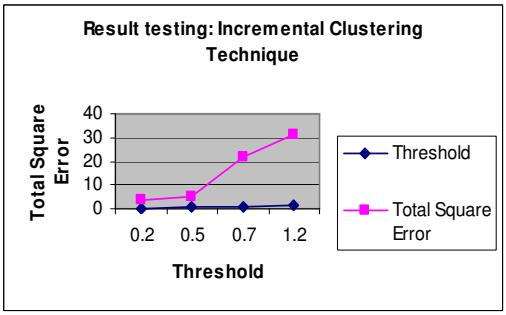

Figure 3. Graph for Incremental Clustering Result

Table 3 and the Figure 3 above shows that when the threshold value increases, the total square error also increased. This is due to the fixed distance between the centroid and data in the cluster becomes bigger.

The result of an experimental for both techniques shows that the partitional technique was depends on the number of cluster to get the lower total square error while as incremental clustering technique depends on the threshold value.

5. CONCLUSION

This paper presented the result of an experimental study of some common clustering techniques. In particular, we compare the two main approaches clustering, partitional and incremental clustering techniques. As a conclusion, partitional clustering outperformed when the number of cluster increased, while the incremental clustering outperformed when the threshold value decreased.

REFERENCES

[1] A. K. Jain, M. N. Murty, and P. J. Flynn, *Data Clustering: A Review*, ACM Computing Surveys, Vol. 31, No. 3, 1999.

[2] Chen, G., Wie Q., Liu, D., and Weets, G. *Simple Association Rule (SAR) and the SAR-based rule discovery*, Computer and Industrial Journal, Vol. 43, Issue 4, 2002, pp 721 – 733.

[3] Clustering at: http://www.eng.man.ac.uk/mech/merg/Research/datafusion.org.uk/techniques/clustering.html. (accessed: 5 February 2005)

[4] Clustering Algorithms at: http://www.cs.uregina.ca/~hamilton/courses/831/notes/clustering/clustering.html. (accessed: 5 February 2005)

[5] Dunham, M. H., *Data Mining: Introductory And Advanced Topics,* New Jersey: Prentice Hall, 2003.

[6] Hoppner, F., Klawonn F., Kruse, R., and Runkler, T., *Fuzzy Cluster Analysis*, John Wiley and Sons, 1999.

[7] Kantardzic, M. *Data Mining: Concepts, Models, Method, And Algorithms,* New Jersey: IEEE Press, 2003.

[8] Steinbach, M., Karypis, G., Kumar, V., *A Comparison of Document Clustering Techniques,* University of Minnesota, Technical Report #00-034, 2000, at http://www.cs.umn.edu/tech_reports/ (accessed: 5 February 2005)