Comparison of the K-Means and the K-Medoids Partitioning Algorithms for Clustering of Optical Mineralogy

Aye Nyein Thu
Computer University (Mandalay)
ayenyeinthu9@gmail.com

Abstract

Clustering is the process of grouping the data into classes or clusters so that objects within a cluster have high similarity in comparison to one another, but are very dissimilar to objects in other clusters. The dissimilarity (or similarity) between the objects described by interval-scaled variables is typically computed based on the distance between each pair of objects. Euclidean distance is used in this system. K-Means and k-Medoids algorithms are by far the most widely used method for discovering clusters in data. This system can implement the clustering of optical mineralogy by using the two partitioning methods (k-Means algorithm and k-Medoids algorithm) and then evaluate the performance of the processing time, squared-error rate and average squared-error value of these algorithms. Experiments show that these two algorithms are effective for 200 records with fourteen attributes of optical mineralogy datasets and becomes more and more effective as the number of clusters increases.

1. Introduction

The major reason that data mining has attracted a great deal of attention in the information industry in recent years is due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge. The information and knowledge gained can be used for applications ranging from business management, production control, and market analysis, to engineering design and science exploration.

Clustering is the process of dividing the data into groups of similar objects according to a similarity measure. The goal is that each group, or cluster, be composed of objects that are similar to each other and dissimilar to objects of other groups. To achieve global optimality in partitioning-based clustering would require the exhaustive enumeration of all of the possible partitions. Instead, most applications adopt one of two popular heuristic methods: (1) the k-Means algorithm, where each cluster is represented by the mean value of the objects in the cluster, and (2) the k-Medoids algorithm, where each cluster is represented by one of the objects located near the center of the cluster [9].

The k-Means algorithm is the most popular among the algorithms mentioned above due to its simplicity and efficiency. However, the k-Medoids based algorithms have been shown to be more robust since they are less sensitive to the existence of outliers, do not present limitations on attribute types (k-Means are restricted to multi-dimensional continuous datasets), and also, because the clustering found does not depend on the input order of the dataset [12].

Optical mineralogy is a science dealing with the study of minerals and their optical properties. Geologists study minerals using a polarized light microscope. The minerals can be determined by knowing more about their compositions. The study of optical mineralogy involves properties of minerals [11]. So, this system intended to develop the Cluster Approach for Optical Mineralogy by using the k-Means and k-Medoids algorithms and then comparison of processing time and squared-error rate of each cluster for these two algorithms.

2. Related work

ProtoMap [7] is designed based on weighted directed graph and CluSTr [6] uses single link method. CLICK [13] uses graph theoretic and statistical techniques and SCOP [3] is based on hierarchical clustering. Krause [1] has used settheoritic method and single linkage clustering for constructing the phylogeny tree of protein sequences. Eui Hong et al. [5] have designed a bypergraph based model using frequent item sets for clustering data. Gurainik et al. [14] have designed a k-Means based algorithm for clustering protein sequences. Eva Bolten et al. [4] have used transitive homol*ogy*, a graph theoritic based approach for clustering. Protein sequences for structure prediction [10].

3. Clustering analysis

Unsupervised data analysis using clustering algorithms provides a useful tool to explore data structures. The aim of clustering methods is to group patterns on the basis of a similarity (or dissimilarity) criteria where groups (or clusters) are set of similar patterns. Crucial aspects in clustering are pattern representation and the similarity measure. Each pattern is usually represented by a set of features of the system under study. It is very important to notice that a good choice of representation of patterns can lead to improvements in clustering performance [12]. The most popular dissimilarity measure for metric representations is the distance, for instance the Euclidean one.

3.1. Interval-scaled variables

Interval-scaled variables are continuous measurements of a roughly linear scale. The measurement unite used can affect the clustering analysis. The dissimilarity (or similarity) between the objects described by interval-scaled variables is typically computed based on the distance between each pair of objects. The most popular distance measure is Euclidean distance**,** which is defined as

$$d(i, j) = \sqrt{\left| x_{i1} - x_{j1} \right|^2 + \left| x_{i2} - x_{j2} \right|^2 + ..x_{ip} - x_{jp} \left.\right|^2}$$

where i = (x_{i1}, x_{i2}, …, x_{ip}) and j =(x_{j1}, x_{j2}, …, x_{jp}) are two p-dimensional data objects [9].

Euclidean distances satisfy the following mathematic requirements of a distance function:

- d (i ,j) >= 0: Distance is a nonnegative number.
- d (i, i) = 0: The distance of an object to itself is 0.
- d (i, j) = d (j, i): Distance is a symmetric function.

d (i, j) <= d (i, h) + d (h, j): Going directly from object i to object j in space is no more than making a detour over any other object h (triangular inequality).

4. Partitioning methods

Given a database of n objects or data tuples, a partitioning method constructs k partitions of the data, where each partition represents a cluster and k <= n. That is, it classifies the data into k groups, which together satisfy the following requirements: (1) each group must contain at least one object, and (2) each object must belong to exactly one group.

It then uses an iterative relocation technique that attempts to improve the partitioning by moving object from one group to another [9]. Most applications adopt one of two popular heuristic methods are the k-Means algorithm and the k-Medoids algorithm.

4.1. K-Means algorithm

The k-Means algorithm [8] is a well known technique for performing clustering on objects in Rn. Each cluster is centred about a point called the centroid, where the centroid's coordinates are the mean of the coordinates of the objects in the cluster. K-Means can be applied to the clustering of rules but there are two reasons for not doing so. Firstly, although each rule may be represented as abinary string, centroids will not have this structure and will make little sense as rules. More importantly, the k-Means algorithm requires that distances to the centroids be calculated for each object at each iteration.

K-Means clustering finds the centroids, where the coordinate of each centroid is the means of the coordinates of the objects in the cluster and assigns every object to the nearest centroid. The algorithm can be described as follows.

Step 1:

Select objects randomly. These objects represent initial group centroids.

Step 2:

Assign each object to the group that has the closest centroid.

Step 3:

When all objects have been assigned, recalculate the positions of the centroids.

Step 4 :

Repeat Steps 2 and 3 until the centroids no longer move Unfortunately, K-Means clustering is sensitive to the outliers and a set of objects closest to a centroid may be empty, in which case centroids cannot be updated [2].

4.2. K-Medoids algorithm

The k-Medoids algorithm, as used to produce the results of this paper. Calculate the mean of the items in each cluster, a representative item, or medoid, is chosen for each cluster at each iteration. The k-Medoids algorithm can be described as follows:

Step 1:

Choose *k* objects at random to be the initial cluster medoids.

Step 2:

Assign each object to the cluster associated with the closest medoid.

Step 3:

Recalculate the positions of the *k* medoids.

Step 4:

Repeat Steps 2 and 3 until the medoids become fixed.

Step 3 could be performed by calculating $\Sigma_{j\in C\ i}\ d(i,\ j)$, for each object *i* from scratch at each iteration. However, many objects remain in the same cluster from one iteration of the algorithm to the next. Improvements in speed can be obtained by adjusting the sums whenever an object leaves or enters a cluster. Step 2 can also be made more efficient in terms of speed, for larger values of *k*. For each object, an array of the other objects, sorted on distance, is maintained. The closest medoid can be found by scanning through this array until a medoid is found, rather than comparing the distance of every medoid [2].

5. Background of the application system

A mineral is a naturally occurring, solid, inorganic element or compound having a uniform composition and a regularly repeating internal structure. Minerals typically have the characteristics of mineral name, chemical composition, crystal system, color, form, cleavage, relief, extinction, twinning, orientation, interference figure, alteration, occurrence and related minerals.

In this system, the user can modify the dataset. This system applied k-Means and k-Medoids algorithms for clustering of the properties of the optical mineralogy training dataset.

K-Means algorithm randomly selects k of the objects, each of which initially represents a cluster mean or cluster. It requires the distances to the centroids be calculated for each object at each iterations. K-Medoids algorithm can be used, which is the most centrally located object in a cluster and performed based on the principle of minimizing the sum of the dissimilarities between each object and its corresponding reference point.

The system can be displayed about the comparison of processing time for these two algorithms and squared-error values. The quality of the resulting clustering is calculated for each such combination. The object is replaced with the object causing the greatest reduction in squared-error.

6. System design

In this system, user can modify training dataset to the system. Training record set form allow researcher to modify imported dataset as necessary such as adding, updating and deleting the data rows.

For cluster process, user needs to enter number of partition (k-value) to the system. And then system can generate k-Means algorithm and k-Medoids algorithm based on user requirement training records set. And also compare the processing time of these two algorithms and calculate the squared-error criteria on each cluster as shown in Figure 1.

In k-Means algorithm portion, accept k value from the user. Define initialize clusters center. The algorithm can process until cluster means no changed. Reassign objects to clusters based on the distance between the objects and the clusters means. All the points are assigned, recalculate the cluster's means.

In k-Medoids algorithm portion, accept k value from the user. Define initialize clusters medoids. The algorithm can process until clusters medoids no changed. Reassign objects to clusters based on the distance between the objects and the clusters medoid. Randomly select non-medoid as $O_{randoms}$. Compute total cost S of swapping current medoid O_{js} and $O_{randoms}$. If $S < 0$, than Swap O_{js} and $O_{randoms.}$

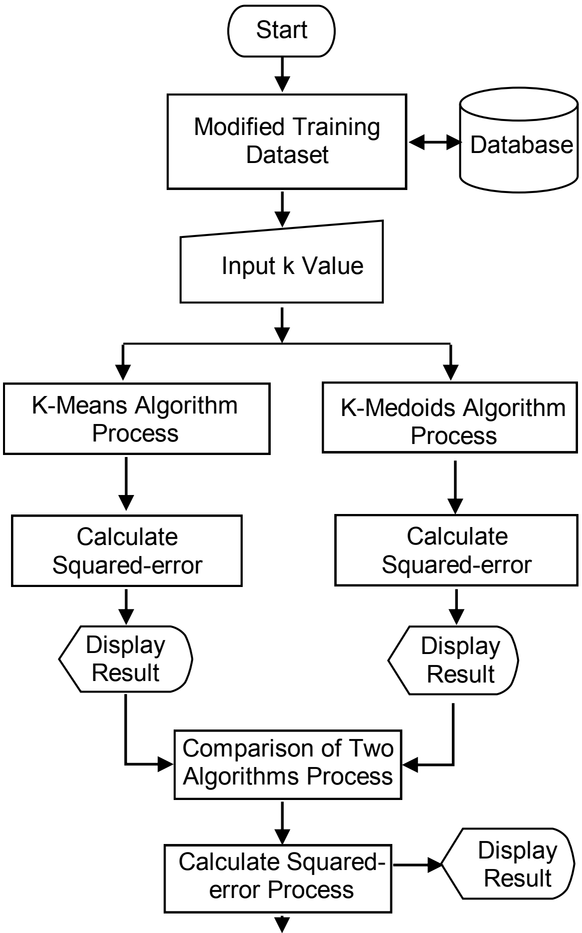
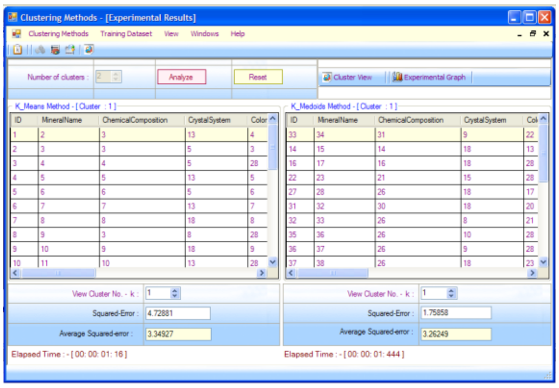

Start
Modified Training Dataset
Database
Input k Value
K-Means Algorithm Process
K-Medoids Algorithm Process
Calculate Squared-error
Calculate Squared-error
Display Result
Display Result
Comparison of Two Algorithms Process
Calculate Squared-error Process
Display Result
Stop

Figure 1. System flow diagram

7. Implementation

This is the multi-documents interface application. This system is created by following controls. They are; menu strip, table layout pane, picture box, label, context menu strip, group box, panel, combo box, button, split container, binding navigator, data grid view and status strip.

In this system, user can select the number of records for generating this system (at least 20 records). So, the system can convert occurrence frequency weighted value for the sample datasets.



K_Means Method - [Cluster : 1]

ID	MineralName	ChemicalComposition	CrystalSystem	Color
1	2	3	13	4
2	3	3	5	3
3	4	4	5	28
4	5	5	13	5
5	6	6	5	6
6	7	7	13	7
7	8	8	18	8
8	9	3	8	28
9	10	9	18	9
10	11	10	13	28

View Cluster No. - k : 1

Squared-Error : 4.72881

Average Squared-error : 3.34927

Elapsed Time : - [00: 00: 01: 16]

K_Medoids Method - [Cluster : 1]

ID	MineralName	ChemicalComposition	CrystalSystem	Col
33	34	31	9	22
14	15	14	18	13
16	17	16	18	28
22	23	21	15	28
27	28	26	18	17
31	32	30	18	20
32	33	26	8	21
35	36	26	10	28
36	37	26	9	28
37	38	26	18	23

View Cluster No. - k : 1

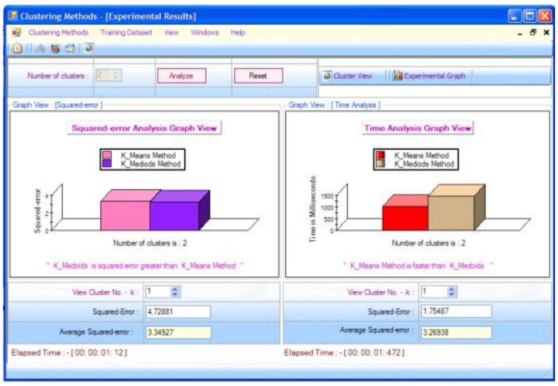
Squared-Error : 1.75858

Average Squared-error : 3.26249

Elapsed Time : - [00: 00: 01: 444]

Figure 2. K-Means and k-Medoids algorithms process form

The user needs to define the number of partition value. After that, the system can generate comparison of the k-Means and k-Medoids algorithm for cluster the optical mineralogy dataset as shown in Figure 2.



Clustering Methods - [Experimental Results]
Clustering Methods
Training Dataset
View
Windows
Help
Number of clusters :
2
Analyze
Reset
Cluster View
Experimental Graph
Graph View : [Squared-error]
Squared-error Analysis Graph View
K_Means Method
K_Mediods Method
Squared-error
0
2
4
Number of clusters is : 2
" K_Medoids is squared-error greater than K_Means Method "
Graph View : [Time Analysis]
Time Analysis Graph View
K_Means Method
K_Mediods Method
Time in Milliseconds
0
500
1000
1500
Number of clusters is : 2
" K_Means Method is faster than K_Medoids "

View Cluster No. - k : 1

Squared-Error : 4.72881

Average Squared-error : 3.34927

Elapsed Time : - [00: 00: 01: 12]

View Cluster No. - k : 1

Squared-Error : 1.75487

Average Squared-error : 3.26938

Elapsed Time : - [00: 00: 01: 472]

Figure 3. Comparison form

User can view the comparing results of processing time, squared-error value and average squared-error values of each cluster by analyzing these two algorithms. User can also view the graph of the comparison results of the k-Means and the k-Medoids algorithm as shown in Figure 3.

7.1. Analysis results

Comparison experimental results of processing time and square error rate for k-Means and k-Medoids algorithms are shown as in Table 1.

Table 1. Comparison results

Algorithms	Data Record	k-value	Processing Time [hr: min: sec: ms]	Cluster Number	Squared-Error Rate	Average Squared-Error Rate
k-Means	120	2	00: 00: 00 :548	1	34.3673	27.1076
				2	19.8479	
k-Medoids	120	2	00: 00: 01 :607	1	34.6158	26.7732
				2	18.9307	

The k-Medoids method is more robust than k-Means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean. However, its processing is more costly than the k-Means method. Both methods require the user to specify k, the number of clusters.

8. Conclusion

Clustering is the division of data into groups of similar objects. The main objective of this system is to find a natural grouping or meaningful partition of optical mineralogy by using a distance or similarity function based on unsupervised learning technique. Data clustering has become an important discovering significant patterns and characteristics large databases. The optical mineralogy using the clustering approach has been shown to be effective in improving the k-Means and k-Medoids based clustering algorithms. The system is to identify the cluster and then to refactor them to aspects, to achieve a system that can be easily understood, maintained and modified.

In this system provided a comparative analysis of two clustering algorithms that are used for identification: k-Means and k-Medoids. For the evaluation have used two quality measures that were previously introduced in the aspect mining literature. In the future plan to apply the clustering algorithms considered in this system to other larger software systems. And also consider to use other unsupervised learning techniques (such as self-organizing maps, Hebbian learning) in order to identify concerns in existing software systems.

9. References

[1] A. Krause, “Large scale Clustering of Protein Sequences”, Ph.D. Dissertation. Berlin, 2002.

[2] A. P. Reynolds, G. Richards, and V. J. Rayward-Smith, “The Application of K-medoids and PAM to the Clustering of Rules”, School of Computing Sciences, University of East Anglia, Norwich

[3] B.L.L.Conte, T.J.P. Ailey, S. Hubbard. A. E.renner. G.Murzin, and C. Chotia, “SCOP: a Structural Classification of Protein Database”, Nucleic Acids Research.28, 2000.

[4] E.Bolten, A.Schliep, and S.Schneckener, “Clustering Protein Sequences-Structure prediction by transitive homology”, GCB, 1999.

[5] E. Hong, G. Karypis, V. Kumar. and B. Mobasher, “Clustering in a High Dimensional Space Using Hypergraph Models”, Research issues on Data Mining and Knowledge Discovery, 1997.

[6] E.V. Kriventseva, W. Fleischmann. E.M. Zdobnov, and G. Apweiler, “CluSTr: a Database of Clusters of SWISSPROT+ TrEMBL Proteins.”, Nucleic Acid,y Resea,nh, 29, 2001.

[7] G. Yona, N. Linial, and M. Linial, “ProtoMap: Automatic Classification of Protein Sequences and Hierarchy of Protein Families:”, Nucleic Acids Research, 28.2000.

[8] J. B. MacQueen. “Some Methods for Classification and Analysis of Multivariate Ob-servations.”, In Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, volume 1, pages 281{297. University of California Press, 1967.

[9] J. Han and M.Kamber, *Data Mining Concepts and Techniques,* ISBN 1-55860-489-8, Morgan Kaufmann Publishers

[10] P. A. Vijaya, M. Narasimha Murty and D. K. Subramanian, “An Efficient Incremental Protein Sequence Clustering Algorithm”, Department of Computer Science and Automation Indian lnstitute of Science, Bangalore - 560012, lndia

[11] P. F. Kerr, *Optical Mineralogy*, Ph.D., Professor of Mineralogy, Columbia University, McGRAW-HILL BOOK COMPANY, INC., New York Toronto London and KOGAKUSHA COMPANY, LTD., Tokyo

[12] P.Ning, T. M.Steinbach and V. Kuma, *Introduction to Data Mining*

[13] R. Sharan, and R. Shamir, “CLICK: A Clustering Algorithm with Applications to Gene Expression Analysis”, Proc. of8‘” ISMB, 2000.

[14] V. Guralnik, and G. Karypis, “A Scalable Algorithm for Clustering Sequential Data”, Proc. o/ is‘IEEE conferenceon Data Mining, 2001.