GRIFFITH UNIVERSITY

An Efficient and Effective Method for Data Mining

Author

Awrangjeb, M, Islam, MM

Published

2001

Conference Title

International Conference on Computer and Information Technology (ICCIT)

Version

Accepted Manuscript (AM)

Rights statement

© 2001 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Downloaded from

http://hdl.handle.net/10072/392037

Link to published version

http://www.iccit.org/

Griffith Research Online

https://research-repository.griffith.edu.au

ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/303812353

An Efficient and Effective Method for Data Mining

Conference Paper · December 2001

CITATIONS
0

READS
26

2 authors:

Mohammad Awrangjeb
Griffith University
80 PUBLICATIONS **1,280** CITATIONS
SEE PROFILE

Mohammad Mahfuzul Islam
Bangladesh University of Engineering and Technology
29 PUBLICATIONS **225** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Special Issue "Edge Detection based on Remote Sensing Data" View project

HAM based data embedment View project

All content following this page was uploaded by Mohammad Awrangjeb on 24 June 2016.

The user has requested enhancement of the downloaded file.

An Efficient and Effective Method for Data Mining

Mohammad Awrangjeb, Mohammad Mahfuzul Islam
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology
Dhaka –1000, Bangladesh.
E-mail: awrangjeb@yahoo.com, mahfuz@cse.buet.edu

Abstract: *Data mining refers to extracting or "mining" knowledge from large amounts of data. It is also called a method of "knowledge presentation" where visualization and knowledge representation techniques are used to present the mined knowledge to the user. Thus, it plays an important role in extracting spatial patterns, features. It also performs presenting data regularity concisely and at higher conceptual levels to recognize spatial databases to accommodate data semantics as well as to achieve better performance, develops a visual feedback querying system to support data mining, it is an essential process where intelligent methods are applied in order to extract data patterns. In this article, a data mining algorithm is explored which is named CLAUMS and proved to be more efficient and effective than existing algorithm PAM. The proposed CLAUMS algorithm is more efficient comparing with respect to both time and memory complexity than PAM.*

Keywords: mining, spatial database, spatial pattern, data semantic, CLAUMS, PAM

1 Introduction

Data mining is called the process of knowledge discovery in databases (KDD) [5,6]. Data mining in general is the search for hidden patterns that may exist in large databases. It is in particular is the discovery of interesting relationships and characteristics that may exist implicitly in spatial databases. Because of the huge amounts (usually, tera-bytes) of spatial data that may be obtained from satellite images, medical equipments, video cameras, etc., it is costly and often unrealistic for users to examine spatial data in detail. Spatial data mining aims to automate such acknowledge discovery process. Thus, it plays an important role in a) extracting interesting spatial patterns and features; b) capturing intrinsic relationships between spatial and non-spatial data; c) presenting data regularity concisely and at higher conceptual levels; and d) helping to reorganize spatial databases to accommodate data semantics, as well as to achieve better performance [3].

Since the 1960s, database and information technology has been evolving systematically from primitive file processing systems to sophisticated and powerful database systems. The research and development in database systems since the 1970s has progressed from early hierarchical and network database systems to the developments of relational database systems (where data are stored in relational table), data modeling tools, and indexing and data organization techniques. In addition, user interfaces optimized query processing, and transaction management. Efficient method for OLAP, where a query is viewed as a read-only transaction, have contributed substantially to the evolution and wide acceptance of relational technology as a major tool for efficient storage, retrieval, and management of large amount of data. Database technology since the mid-1980s has been characterized by the popular adoption of relational technology and an upsurge of research and development activities on new and powerful database system.

However, the algorithms developed so far suffer from the following problems: First, the user or an expert must provide the algorithms with spatial concept hierarchies, which may not be available in many applications. Second, the algorithms conduct their spatial exploration primarily by merging regions at a certain level of the hierarchy to a larger region at a higher level. Thus, the quality of the results produced by the algorithms relies quite crucially on the appropriateness of the hierarchy to the given data. Discovering this hierarchy may itself be one of the reasons to apply spatial data mining.

To deal with these problems, whether cluster analysis techniques are applicable is explored here. Cluster Analysis is a branch of statistics that in the past three decades has been intensely studied and successfully applied to many applications. To the spatial data mining task at hand, the attractiveness of cluster analysis is its ability to find structures or clusters directly from the given data, without relying on any hierarchies. However, cluster analysis has been applied rather unsuccessfully in the past to general data mining and machine learning. The complaints are that cluster analysis algorithms are ineffective and inefficient. Indeed, for cluster analysis algorithms to work effectively there need to be a natural notion of similarities among the "objects" to be clustered. And traditional cluster analysis algorithms are not designed for large data sets, say more than 2000 objects [3].

For data mining, the approach here is to apply cluster analysis only to the spatial attributes, for which natural notions of similarities exist. As will be shown in this paper, in this way, cluster analysis techniques are effective for data mining. As for the efficiency concern, CLAUMS, which is designed for larger data sets and is more effective and efficient than PAM [2] that is shown here. Moreover, in comparison with

respect to time and space complexity it is better than existing PAM, CLARA and CLARANS [11]. Because PAM takes $O(k(n-k)^2)$ for each iteration [3], CLARA takes $O(ks^2+k(n-k))$ for each iteration [1], and CLARANS takes $O(n^2)$ [1]; whereas our proposed algorithm takes $O(n+2kn)$ ie. linear time for executing the whole algorithm. Where s is the size of the sample, k is the number of clusters, n is the total number of objects. Again the proposed algorithm CLAUMS only takes a less memory space in comparison with existing algorithms that is shown in this paper. More specifically, followings will be reported in this paper:

- Existing well-known algorithm in cluster analysis, which is called PAM. It is one of the first *k*-medoids algorithms introduced.
- Time and space complexity analysis for PAM.
- The proposed algorithm CLAUMS is described next. It is based on the medians of samples.
- Time and space complexity for CLAUMS.
- Comparisons between PAM and CLAUMS.

2 PAM (Partitioning Around Medoids): A Clustering Algorithm Based on k-Medoids

In the past 30 years, cluster analysis has been widely applied to many areas such as medicine (classification of diseases), chemistry (grouping of compounds), social studies (classification of statistical findings), and so on. Its main goal is to identify structures or clusters that present in the data. While there is no general definition of a cluster, algorithms have been developed to find several kinds of clusters: spherical, linear, drawn out, etc. See [2] and [4] for more detailed discussions and analysis of these issues. In this section, PAM, the best known k-medoid method on which our algorithm is based is presented. PAM (Partitioning Around Medoids) was developed by Kaufman and Rousseeuw [2]. To find k clusters, PAM's approach is to determine a representative object for each cluster. This representative object, called a medoid, is meant to be the most centrally located object within the cluster. Once the medoids have been selected, each non-selected object is grouped with the medoid to which it is the most similar.

More precisely, if O_j is a non-selected object, and O_i is a medoid (selected), we say that O_j belongs to the cluster represented by O_i, if $d(O_j,O_i) = \min_{Oe} d(O_j,O_e)$, where the notation $\min_{Oe}$ denotes the minimum over all medoids O_e, and the notation $d(O_{a,}O_b)$ denotes the dissimilarity or distance between objects O_a and O_b. All the dissimilarity values are given as inputs to PAM. Finally, the quality of a clustering (i.e. the combined quality of the chosen medoids) is measured by the average dissimilarity between an object and the medoid of its cluster. To find the k medoids, PAM begins with an arbitrary selection of k objects. Then in each step, a swap between a selected object O_i and a non-selected object O_h is made, as long as such a swap would result in an improvement of the quality of the clustering. In particular, to calculate the effect of such a swap between O_i and O_h , PAM computes costs C_{jih} for all non selected objects O_j. Depending on which of the following cases O_j is in, C_{jih} is defined by one of the equations below [3].

- First Case: suppose O_j currently belongs to the cluster represented by O_i. Furthermore, let O_j be more similar to $O_{j,2}$ than O_h, i.e. $d(O_j,O_h) \geq d(O_j,O_{j,2})$, where $O_{j,2}$ is the second most similar medoid to O_j. Thus, if O_i is replaced by O_h as a medoid, O_j would belong to the cluster represented by $O_{j,2}$. Hence, the cost of the swap as far as O_j is concerned is:

  $$\mathbf{C_{jih} = d(O_j,O_{j,2}) - d(O_j,O_i)} \quad \mathbf{.....(1)}$$

  This equation always gives a non-negative C_{jih}, indicating that there is a non-negative cost incurred in replacing O_i with O_h.

- Second Case: O_j currently belongs to the cluster represented by O_i. But this time, O_j is less similar to $O_{j,2}$ than O_h, i.e. $d(O_j,O_h) < d(Oj,Oj_{,2})$. Then, if O_i is replaced by O_h, O_j would belong to the cluster represented by O_h. Thus, the cost for O_j is given by:

  $$\mathbf{C_{jih} = d(O_j,O_h) - d(O_j,O_i)} \quad \mathbf{.....(2)}$$

  Unlike in Equation (1), C_{jih} here can be positive or negative, depending on whether O_j is more similar to O_i or to O_h .

- Third Case: suppose that O_j currently belongs to a cluster other than the one represented by O_i. Let $O_{j,2}$ be the representative object of that cluster. Furthermore, let O_j be more similar to $O_{j,\ 2}$ than O_h. Then even if O_i is replaced by O_h, O_j would stay in the cluster represented by $O_{j,2}$. Thus, the cost is:

  $$\mathbf{C_{jih} = 0} \quad \mathbf{.....(3)}$$

- Fourth Case: O_j currently belongs to the cluster represented by $O_{j,2}$. But O_j is less similar to $O_{j,2}$ than O_h. Then replacing O_i with O_h would cause O_j to jump to the cluster of O_h from that of $O_{j,2}$. Thus, the cost is:

  $$\mathbf{C_{jih} = d(O_j,O_h) - d(O_j,O_{j,2})} \quad \mathbf{....(4)}$$

  and is always negative.

Combining the four cases above, the total cost of replacing O_i with O_h is given by:

$$\mathbf{TC_{ih} = \Sigma C_{jih}} \quad \mathbf{.....(5)}$$

Here summation is over all O_j that are not currently selected. The Algorithm PAM is presented below.

Algorithm PAM

1. Select k representative objects (Medoids) arbitrarily.
2. Compute TC_{ih} for all pairs of objects O_i,O_h where O_i is currently selected, and O_h is not.
3. Select the pair O_i,O_h which corresponds to $min_{Oi,Oh}$ TC_{ih}.. If the minimum TC_{ih} is negative, replace O_i with O_h, and go back to Step 2.
4. Otherwise, for each non-selected object, find the most similar representative object. ***Halt****.*

Time & Space Complexity for PAM

Experimental results show that PAM works satisfactorily for small data sets (e.g. 100 objects in 5 clusters). But it is not efficient in dealing with medium and large data sets. This is not too surprising if we perform a complexity analysis on PAM. In Steps 2 and 3, there are altogether k(n-k) pairs of O_i,O_h. For each pair, computing TC_{ih} requires the examination of (n-k) non-selected objects. Thus, Steps 2 and 3 combined is of $O(k(n-k)^2)$. And this is the complexity for only one iteration. Thus, it is obvious that PAM becomes too costly for large values of n and k. So total time complexity is[3]:

O(number of iterations $k(n-k)^2$).*

When the space complexity for PAM is considered the situation causes more headaches. For n objects there will be need *n*sizeof(object)* byes; there is also a temporary two dimensional array of size *(n-k)(n-k+1)*sizeof(integer)* bytes keeping track of sources and destinations index if O_i would be replaced by O_h; and another two dimensional array of size *(k+1)(n-k+1)*sizeof(integer)* bytes (if distances are taken as integer valued) keeping track of current distances between each pair O_i, O_h. So total space complexity is:

*O(n*sizeof(object) + (n-k)(n-k+1)* sizeof(integer) + (k+1)(n-k+1)* sozeof(integer))* bytes.

This analysis motivates the development of CLARA and CLARANS [3].

3 The Proposed CLAUMS Algorithm

Clustering in data mining [7,8] is a discovery process that groups a set of data such that the intracluster similarity is maximized and the intercluster similarity is minimized [2,8,9,10]. The proposed algorithm CLAUMS (Clustering Larger Applications Using Medians of Samples) algorithm is based on the thoughts that the objects that reside at the middle in any sorted object list (sample) contain themselves the medoid for the object list (sample). This medoid represents the objects in the list (sample) with minimum (optimum) distance in most cases. So, first the objects in the database are sorted. For sorting the efficient sorting algorithm such as counting sort that runs in linear time is used. Then the sorted object list $O=\{O_1,O_2,\ldots\ldots\ldots..,O_n\}$ is broken into k consecutive samples $S_i=\{O_j,O_{j+1},\ldots\ldots\ldots\ldots\ldots,O_m\}$; where i = 1,2,……,k; for successive samples S_p and S_q if O_s be the last object of sample S_p and O_t be the first object of sample S_q then t=s+1; j=1,2,…….n; m-j+1= SampleLength, for all samples except for the sample S_k, the last sample, for which m-j+1≤ SampleLength, where SampleLength=n/k.

First, the list is sorted using counting sort algorithm. Second, the median of a sample and its index $index_{med}$ in the sorted list is determined. For each O_j, where j = ($index_{med}$–k) to ($index_{med}$+k-1), and each O_i in the sample the distance d(O_j,O_i) is found and the total cost equation here

$$TC_{Oj} = \Sigma\, d(O_j,O_i) \text{ -------(a)}$$

where i takes index values of objects in that sample. Third, the Object O_j is taken as medoid for that sample for which TC_{Oj} is minimum. The Algorithm CLAUMS is given below.

Algorithm CLAUMS

1. SampleLength=n/k.
2. Sort the object list using counting sort algorithm.
Get a non-examined sample S find the median O_{med} of S; find the index of O_{med} in sorted object list; let the index be $index_{med}$.
3.For j = $index_{med}$–k to $index_{med}$+k-1 find TC_{Oj} using equation (a).
4.Find the object O_j for which TC_{Oj} is minimum. It is the medoid for that sample.
5.If there is any sample not examined go to step 3. Else ***Halt****.*

Time & Space Complexity for CLAUMS

The time complexity for counting sort algorithm in step 2 is *O(n)*. For each iteration from step 3 to 5 is *O(2k*SamleLength)*. There are total k iterations. So, total time complexity for CLAUMS is $O(n+2k^2*SampleLength)\approx O(n+2nk)$; since SampleLength=n/k. Which is linear if k is kept fixed.

To find memory complexity, *n*sizeof(Object)* bytes is needed like PAM for the object list; to keep distances in step 4 takes *SampleLength*sizeof(integer)* bytes (if distances are taken as integer valued). So total space complexity for CLAUMS is

*O(n*sizeof(Object)+n*sizeof(integer)/k)* bytes;

since SampleLength = n/k.

4 Results Found

In theoretical the following time and space complexity values for PAM and CLAUMS are found. The time complexity for PAM is given for each iteration, whereas, for CLAUMS it is for executing whole

algorithm. Here the assumption is that the sizeof(object) is 2 bytes.

n	k	Time Complexity		Space Complexity	
		PAM	CLAUMS	PAM	CLAUMS
20	2	648	100	838	60
50	5	10125	550	4792	120
100	5	45125	1100	19592	240
100	10	81000	2100	18582	220
200	10	361000	4200	77182	440
500	10	2401000	10500	492982	1100
500	20	4608000	20500	482962	1050
1000	10	9801000	21000	1985982	2200
1000	20	19208000	41000	1965962	2100
1500	10	22201000	31500	4478982	3300
1500	20	43808000	61500	4448962	3150
2000	10	39601000	42000	7971982	4400
2000	20	78408000	82000	7931962	4200
2000	25	97515625	102000	7911952	4160

Table 1

Next the time complexity values that are found practically for various n and k are given. Here the sizeof(object) is 2 as theoretical. It should be mentioned here that Intel Pentium S PC with 133MHz speed and 16MB RAM and no external cache memory is used for simulation , since time depends on these configurations. Table 2 represents distances in PAM and CLAUMS. For higher values of n PAM does not give satisfactory results. So, results for higher n values are not included here.

n	k	Object Range	PAM	CLAUMS
12	3	12	15	12
50	5	50	544	108
100	10	200	615	488
120	10	220	2160	647

Table 2

Table 3 represents running time vs k for various n for CLAUMS.

	Medoids k								
n	10	15	20	25	30	35	40	45	50
1000	55	55	55	55	55	55	110	110	110
2000	55	55	110	110	110	110	165	165	220
3000	55	110	165	165	165	220	275	275	330
4000	110	165	220	275	275	330	385	440	495
5000	165	220	275	330	385	440	495	605	659
6000	220	275	330	385	495	549	659	769	824
7000	275	330	385	440	604	659	769	934	989
8000	330	385	440	495	714	769	879	1099	1154
9000	385	440	495	550	824	879	989	1264	1317
10000	440	495	550	604	934	989	1100	1429	1481

Table 3

Table 4 represents running time vs k for various n for PAM.

k	n=70	n=80	n=100	n=110
	Time(ms)	Time(ms)	Time(ms)	Time(ms)
2	0	0	0	0
5	54.945	54.945	55	109.819
10	94.835	119.89	149.89	219.783
14	111.095	156.78	196.64	274.725
20	164.835	204.65	239.835	335.352

Table 4

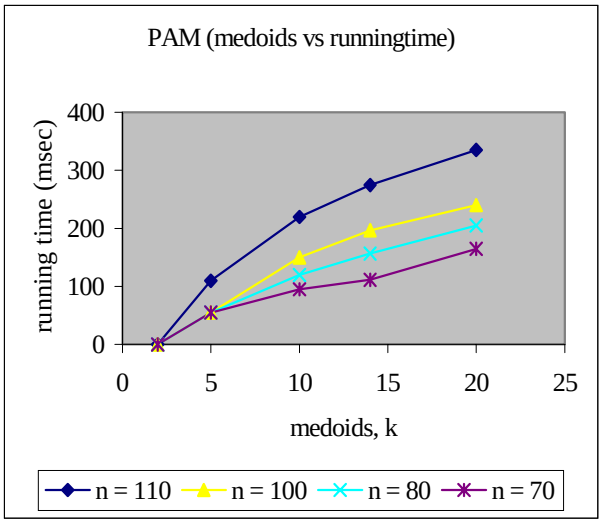
5 Comparing PAM and CLAUMS

For small values of n (because PAM does not give satisfactory results for large n) CLAUMS always gives clusters with less cost (distance) than PAM that is observed from Table 2.

Time for PAM is not given in table 3 because time found to be dissatisfactory for large n (e.g. for n>110) in the case of PAM algorithm. Again, time for CLAUMS is not given in table 4 because time found to be zero for small n (e.g. for n<500) in the case of CLAUMS algorithm.

So, PAM and CLAUMS are compared individually with respect to time.

Graph 1 represents k vs running time for various n for PAM. Graph 1 is drawn using results in table 4. Graph 2 represents k vs running time for various n for CLAUMS. Graph 2 is drawn using results in table 3. Graph 3 represents n vs running time for various k for CLAUMS. Graph 3 is drawn using results in table 3.

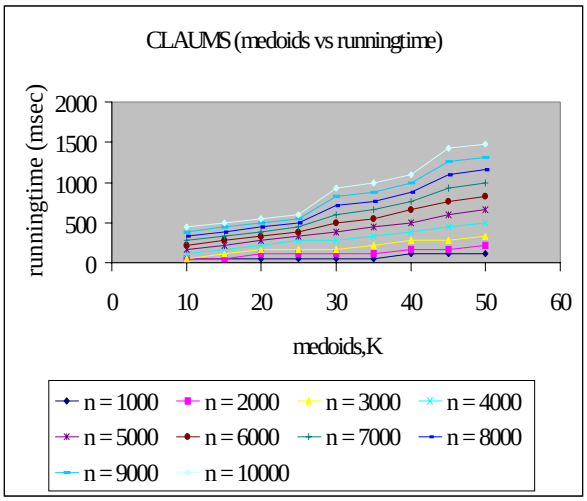


Graph1

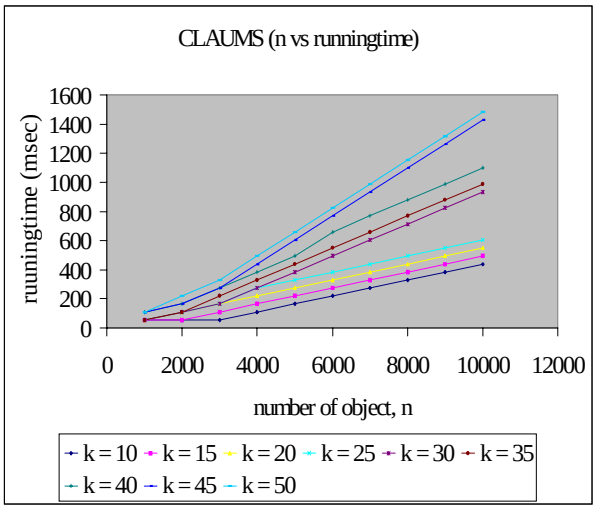
6 Future Works

Since both CLARA and CLARANS have time complexity $\theta(n^2)$ on the average, the proposed

algorithm must give better results than CLARA and CLARANS. Again, CLARA and CLARANS do not give optimal clusters. So we say that CLAUMS will give optimal clustering with minimum cost, which we want to prove in future.



Graph2



Graph3

7 Conclusion

CLAUMS takes linear time complexity for fixed k and requires a memory complexity that is very small in comparison with other existing algorithms require.

References:

[1] Jiawei Han, Micheline Kamber. *Data Mining: Concepts and Technique.* Editor: Diane D. Cerra, Academic Press, A Harcourt Science and Technology, 2001, San Diego, USA, page 324. http://www.academicpress.com.

[2] L. Kaufman, P.J.Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis,* John Wiley & Sons, 1990.

[3] Raymond T. Ng, Jiawei Han. *Efficient and Effective Clustering Methods for Spatial Data Mining,* VLDB Conference, Santiago, Chile, 1994.

[4] H. Spath. *Cluster Discussion and Analysis: Theory, FORTRAN Programs, Examples,* Ellis Horwood Ltd. 1984.

[5] Usama M. Fayad, Gregory Piatetsky-Shapiro and Padhraic Smyth. *Advances in Knowledge Discovery and Data Mining.* AAAI Press / The MIT Press, 1996, American Association of Artificial Intelligence, California, USA. Page 35.

[6] W. J. Frawley, Gregory Piatetsky-Shapiro and C. J. Matheus. *Knowledge Discovery in Databases.* AAAI Press / The MIT Press, 1991. Page 1-27.

[7] M. Stonebraker, R. Agrawal, U. Dayal, E. J. Neuhold, and A, Reuter. *DBMS research at a crossroads: The Vienna update.* In Proc. Of 19th VLDB conference, Dublin, Ireland, 1993. Pages 688-692.

[8] M. S. Chen, J. Han and P. S. Yu. *Data Mining: An Overview from database perspective.* IEEE Transactions on Knowledge and Data Eng., December 1996. Pages 866-883,

[9] A. K. Jain and R. C. Dubes. *Algorithm for Clustering Data*. Prentice Hall, 1988.

[10] L. J. Hubert, P Arabie and G. De Soete. *Clustering and Classification.* World Scientific, 1996.

[11] R. Ng and J. Han. *Efficient and Effective Clustering Method for Spatial Data Mining.* In Proc. Of the 20th VLDB Conference, San Diego, CA, 1995.

View publication stats