International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:268
Vol.2, No.6, June 2013 DOI:10.15693/ijaist/2013.v2i6.20-24

# Clustering Algorithms In Datamining

K. Sheik Ibrahim
M.E Computer& communication
Sethu Institute of Technology

M. Mathina Kani
Assistant Professor (sr.g)
Sethu Institute of Technology

**Abstract---GNG query is a partition-based clustering problem which can be found in many real applications and is NP hard .Given a data point set D, a query point set Q, and an integer k, the Group Nearest Group (GNG) query finds a subset (ω(|ω|≤ $K$) of points from D such that the total distance from all points in Q to the nearest point in ω is not greater than any other subset ώ(|ώ| ≤ $K$) of points in D. Exhaustive Hierarchical Combination (EHC) algorithm and Subset Hierarchical Refinement (SHR) algorithm are developed for GNG query processing. While EHC is capable to provide the optimal solution for k = 2, SHR is an efficient approximate approach that combines database techniques with local search heuristic. The processing focus of our approaches is on minimizing the access and evaluation ofsubsets of cardinality k in D, since the number of such subsets is exponentially greater than |D|. To do that, the hierarchical blocks of data points at high level are second-hand to find an transitional solution and then refined by subsequent the guided search direction at low level so as to prune irrelevant subsets. The all-inclusive experiments on both real and synthetic data sets demonstrate the superiority of SHR in terms of efficiency and quality.**

**KeyTerms : Group nearest group (GNG), K-median clustering, Group nearest neighbour (GNN).**

## 1. INTRODUCTION

Data mining is the process of semi-automatically analyzing large databases to find patterns that are valid, novel, useful, understandable. It is also known as Knowledge Discovery in Databases (KDD).Data mining is a hot buzzword for a class of techniques that find patterns in data. It is a user-centric, interactive process which leverages analysis technologies and computing power. It is a group of techniques that find relationships that have not previously been discovered. It is not reliant on an existing database. Data mining overlaps with machine learning, statistics, artificial intelligence, databases, visualization.GNG query is a partition based clustering which can be found in many applications. The GNG query can be defined in a formal way. Given a set of data points D, a query point set Q, and an integer value k ($1 \le k \le \min\{|D|, |Q|\}$), GNG query finds a subset of points from ώ from D ($|ώ| \le k$) for every ώ € P(D) ($|ώ| \le k$), where P(D) is the power set of D. A GNG query may return k points. In some cases, a GNG query may return a solution with less than k points. No matter what the setting of k is, the minimal total traveling distance is the distance from all persons. If we are aware of the cardinality of the solution k', processing GNG query is to find the subset of k' points that produces the minimum total traveling distance. k' is not known in advance when processing GNG query. When k = 1, a GNG query reduces to a GNN query which returns an individual point in D. The focus of GNN query processing is on minimizing access and evaluation of data points in D. GNG query, also known as k-median clustering in operations research, is a partition-based clustering problem which groups data points into k (a given number) clusters based on an optimization objective function. The partition-based clustering problem is NP-hard .The k median clustering has been investigated using local search heuristics which are popular for hard combinatorial optimization problem. The typical process of the local search heuristics finds an arbitrary solution and iteratively optimizes it with the operation of swap (i.e., replacing a point in the solution with another point in the search space) till improvement can be obtained. In practice, using local search heuristics for GNG query leads to a gap of a few percentage points between the obtained solution and the global optimum. Database techniques are explored to boost the GNG query processing of local search heuristics without any loss on clustering quality.

## 2.EXHAUSTIVE HIERARCHICAL COMBINATION (EHC) ALGORITHM

EHC aims to find the optimal solution without evaluating all subsets of k points. In EHC, every set of k blocks is evaluated in high hierarchical level and the set with the current best value (i.e., the minimum total distance) are refined by visiting their children in next level. EHC is capable to provide the optimal solution. However,

the performance of EHC drops down dramatically when k increases from 2.

In EHC, a GNG query is processed by traversing the R-tree of D. At the root, every k entry forms a subset denoted as ω. For a query point q € Q and a subset ω, the minimum distance from q to an entry E in ω (E is the minimum bounding rectangle of data points under E in R-tree is denoted as function mindist(q,E) and the maximum distance is denoted as function maxdist(q,E). Given a set Q of query points and a subset ω, the query points can be grouped to the closest entries in ω according to these two distance functions.

For distance function mindist(q,E), the distance sum from query points to the closest entries in ω is denoted as $\sum_{q\in Q}$mindist(q, ω), and $\sum_{q\in Q}$maxdist(q, ω) for distance function maxdist(q,E), that is, ω has two distance sums. Let a leaf node be the child leaf node of ω if it is under one entry in ω. It is clear that any subset formed by k child leaf nodes of ω must have a distance sum in between $\sum_{q\in Q}$mindist(q, ω) and $\sum_{q\in Q}$maxdist(q, ω). Thus, we denote $\sum_{q\in Q}$mindist(q, ω) as $\sum_{lb}$and $\sum_{q\in Q}$maxdist(q, ω) as $\sum_{ub}$.At the root, let $\{\omega_1,\ldots,\omega_n\}$ be all possible subsets of k entries. When each of these subsets computes $\sum_{lb}$ and$\sum_{ub}$ , they are inserted into a heap H. The minimum $\sum_{ub}$ in H is assigned to a threshold £. If a subset ω in H has lower bound distance $\sum_{lb}^{\omega}\geq$ £, it can be safely pruned from H. That is, the child leaf nodes of ω cannot form a subset of size k with a distance sum less than £. For the remaining subsets in H, following the best first traverse fashion, the ω with the minimum $\sum_{lb}$ is selected, the direct child nodes of ω (i.e.,the nodes directly referred by entries in ω) are visited and every k entries of these child nodes form new subsets. The $\sum_{lb}$ and$\sum_{ub}$ are computed for each of these new subsets.

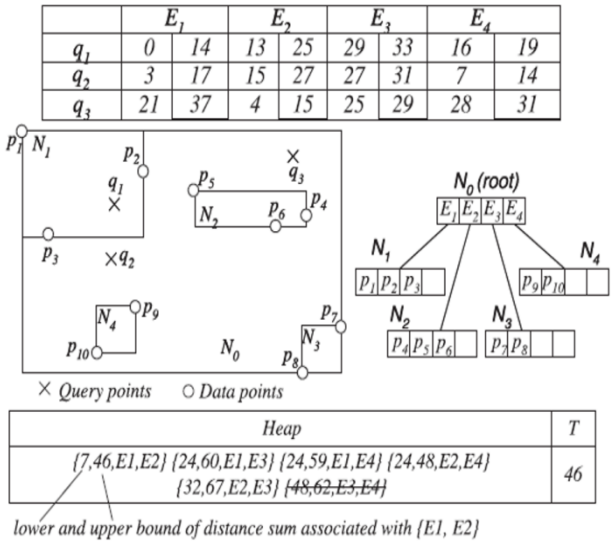


Fig 1: Query Processing when k = 2

After inserting them into H, the threshold £ is set as the current minimum$\sum_{ub}$ in H. Similarly, any subset in H with$\sum_{lb}\geq$£ can be safely pruned. These operations are repeated until the minimum$\sum_{lb}$ in H is associated with a subset of leaf nodes (in this case $\sum_{lb}=\sum_{ub}$ ). This subset is returned by EHC. For description clarity, the subsets are inserted into the heap and then pruned. However, the pruning can also take place before inserting subsets into H. Fig.1 shows an example of EHC. The entries in the root are combined into subsets of size k = 2 and these subsets are inserted into a heap. The minimum $\sum_{ub}$ in H is 46. Using£ = 46 as the threshold, the subset{E3, E4}can be pruned from the heap. Then, {E1,E2} are replaced by all possible subsets of size 2 composed by direct child nodes of E1, E2.

## 3. SUBSET HEIRARCHIAL REFINEMENT (SHR) ALGORITHM

EHC is workable only for small k even though optimization is applied. In this section, we describe the Subset Hierarchial Refinement algorithm. It can provide a high quality solution while the performance is linear with k, D, and Q. The time complexity is O(k(|D| - k)|Q|). SHR is inspired by PAM. SHR works basically as follows: first, an initial subset ω is identified; then, SHR tries to refine ω by tentatively replacing elements in ω by points in D - ω until the total distance cannot be reduced further.

### 3.1 PRUNING RULES

Before going to the details of SHR, we first propose two pruning rules which explore the proximity of the query points in order to prune the irrelevant subsets.

**INITIAL CANDIDATE PRUNING (ICP)**

Given D, Q, and a subset ω- D, we tentatively use points in D -ω to replace the points in ω in order to obtain an updated ω which produces a smaller total distance. During this process, a point p' €D -ω can be pruned if d(q,p')≥d(q,ω) for q € Q.ICP aims to reduce the search space when optimizing a given ω by replacing one point p €ω with a data point p' € D -ω. The replacement should reduce the total distance from query points in Q to the closest points in ω.If a point p' € D -ω cannot reduce the total distance by replacing it with any point in ω, p' is not relevant to the optimization of ω. The purpose of ICP is to identify such p' and prune it. ICP is valid when p' is an entry of a node in R-tree. When a node N is visited during the tree browsing, the entries in N are processed. To each entry, say E, the distance mindist(q,E) is computed from all q € Q. E can be pruned if mindist(q,E)- d(q,ω) for q € Q.

The reason is that replacing any p €ω by any data point under E in Rtree cannot reduce the total distance.

**FURTHER CANDIDATE PRUNING (FCP)**

Given D, Q, and a subset ω- D, we tentatively use points in D -ω to replace a point p €ω in order to obtain an updated ω which produces a smaller total distance. During this process, p' €D -ω can be pruned if the resultant total distance cannot be reduced. If a point p'€ D -ω is not pruned by ICP, FCP is applied to check whether replacing p' with some points in ω can be avoided. The idea is to compute the total distance when replacing p' with a point in ω. Clearly, if p' is a data point, every possible replacement is processed and thus no computation is avoided.

The pruning power of FCP takes effect when p' is an entry of an R-tree node. As shown in Fig.1 , when replacing p3 by the entry E4, q1, q 2 will change to E4 and q3 is still to p4. Since the resultant total distance cannot be reduced, any data point under E4 cannot diminish the total distance if it replaces p3 in ω. Thus, the replacement between E4 and p3 can be pruned safely and so does E4 and p4.Both FCP and ICP are used in SHR when browsing Rtree. When a node is visited, the entries in this node are pruned using ICP first, and then the remaining entries are pruned using FCP. The entries pruned by ICP can also be pruned by FCP. That is, FCP does decide the number of entries accessed and processed. The purpose of ICP is that it is cheaper to prune some entries and so FCP is avoided to do that same job. The effect of ICP to performance is stronger when query points are distributed in a small region.

## 4. PARTITION-BASED CLUSTERING

Given an integer k and a data point set D, database clustering is the process of grouping data points together based on a user defined similarity function. Data points are similar to each other when they are in the same cluster and dissimilar when they are in different clusters. Two well known partition-based algorithms are the k-means and k medoids. The k-means algorithm uses the centroid (a virtual point rather than a data point) of the objects in each cluster as the cluster center while k-medoids algorithm uses the most centrally located point as the center. The overall distance for data points to the associated cluster centers is minimum. The k-median clustering is the bichromatic version of k-medoids clustering where there are two datasets, and one data set is grouped to k cluster centers from the other data set. In general, the task of finding a global optimal k partition belongs to the class of NP-hard problems. For this reason, heuristics are used to achieve a balance between the efficiency and effectiveness close to the global optimum as much as possible. The algorithm for k-means clustering is relatively scalable and efficient in processing large datasets because the computational complexity of the algorithm is O (nkt), where n is the total number of objects, k is the number of clusters, and t is the number of iterations. Normally, k << n and t <<n. The method often terminates at a local optimum. The k-means is very sensitive to noise and outlier data points since a small number of such data can substantially influence the mean value. K-medoids/median clustering is less sensitive to noise and outliers. However, this results in a higher running time.

## 5. LOCAL SEARCH HEURISTICS

The local search heuristic is popular in clusterings of k partitions and this process includes two steps . The first step arbitrarily selects k points as the initial cluster centers. The data points are grouped to the nearest center and cluster quality is computed. In the second step, the iterative relocation technique is adopted. In each iteration, some/all cluster centers are updated and the cluster quality is recomputed. The second step terminates when the cluster quality cannot be further improved. A well-known local search heuristic for k-mediod/median clustering is PAM . Using PAM, k centers are randomly selected in the first step to form the initial set of medoids/median, say ω. In each iteration of the second step, PAM performs following operations: tentatively replacing a current medoid/median p € ω by a point p'= D -ω , the cluster quality improvement is computed, e.g., the reduction of the distance sum; among all such tentative replacements, the one with the maximum improvement is carried out. PAM repeats this operation until the cluster quality cannot be improved any further. The solution of k-medoids/k-median clustering can be solved with local search heuristics within a factor of at most 5 from the global optimum.

## 6. GNG QUERY PROCESSING

The query processing starts by selecting an initial subset $\omega_{ini}$. We can arbitrarily select k points from D as $\omega_{ini}$. However, if $\omega_{ini}$ produces a smaller total distance, fewer replacements are required to reach the final solution and thus a better overall performance. In this work, we employ an approach based on k-means algorithm which considers the distribution of both Q and D. First, Q is clustered using k-means algorithm. Then, for each obtained center (i.e.,mean), the nearest neighbor from D is retrieved as a member of $\omega_{ini}$ until there are k distinct points in $\omega_{ini}$. A noteworthy point is that $\omega_{ini}$ is not reliable to be an approximate solution of GNG. At the root N for the first iteration, we tentatively replace Each p € ω cur by every entry E € N and compute the resultant total distance sum. For an entry E and a point p €$\omega_{cur}$, only if the resultant sum <£, we record{sum, p,E} in a heap H (empty initially); otherwise, this replacement can be safely pruned according

to the pruning rules. After processing tentative replacements between all entries in N and all points in $\omega_{cur}$, if H is empty, SHR terminates by returning $\omega_{cur}$ since no replacement can further reduce the total distance; else if H is not empty, the element h € H with the minimum sum is removed from H for further processing. For h with h:E referring to a non leaf node, the entries in this node are processed in the similar way as we process the root.

The only difference is that these entries only tentatively replace with h rather than all p € $\omega_{cur}$. For h with h:E referring to a leaf node, say p', the first iteration is done by replacing h:p with p' in $\omega_{cur}$ and resetting £=h.sum. In the next iteration, the R-tree is browsed again with the updated $\omega_{cur}$ and £ in the same way. SHR terminates until no further reduction of the total distance is possible as discussed above. The R-tree is browsed by visiting the root first. Replacing each point in the initial subset with every entry in the root, the resultant subset and the associated total distance are recorded in a heap. If the total distance is greater than £, the resultant subset can be safely pruned.

Two subsets {E1, p4, 10}{p3, E2 ,21} remain in the heap. Since {E1, p4, 10} has the minimum total distance, E1 is replaced by every entry in node N1 in this subset. Subsequently, the subset {p2; p4; 22 } has the minimum otal distance. Since p2,p4 are leaf nodes, the first iteration stops; the current subset and £are updated to{p2 ,p4,22} and 22.As a further step to cut off the tentative replacements, £can be updated during the R-tree traversed within one iteration. It is similar to the method used in EHC. When tentatively replacing a point p € $\omega_{cur}$ with an entry E, we can also group query points based on their maximum distances to E. The consequential total distance is the upperbound of the total distance of the subset which is obtained by replacing p with any data point under E in R-tree. It is denoted as $\sum_{ub}$.If$\sum_{ub}$< £, £ is updated to$\sum_{ub}$ . For an element h € H, it can be pruned safely pruned if h.sum≥£.

## 7. ANALYSIS OF SHR VERSUS PAM AND CLARANS

k-medoids algorithms PAM and CLARANS can be used to process GNG queries. Using PAM, k centers are randomly selected in the first step to form the initial set of medoids, say ω. In each iteration of the second step, PAM performs following operations. Tentatively replacing each current medoid p € ω by every point p'= D - ω , the cluster quality improvement is computed, e.g., the reduction of the distance sum. Among all such tentative replacements, the one with the maximum improvement is carried out. In next iteration, PAM repeats this operation until the cluster quality cannot be improved any further. The complexity of PAM in one iteration is O (k (|D|- k)$^2$ ). For a large D, such computation becomes costly. If the initial subset is the same, SHR and PAM return the identical solution since SHR is equivalent to PAM without the proposed database techniques to improve the processing efficiency. Compared to PAM, SHR is more efficient as it avoids unnecessary tentative replacements with support of R-tree and pruning rules. PAM can always find solutions of GNG query that are within a factor of at most 5 from the optimum and for practical, non-pathological instances, the gap is usually much smaller, just a few percentage points. In CLARANS, one randomly selected point in $\omega_{cur}$ is tentatively replaced by one randomly selected point in D -$\omega_{cur}$. If the total distance can be reduced, $\omega_{cur}$ is updated by carrying out this replacement immediately. For the updated $\omega_{cur}$, if no better solution is found after maxneighbor attempts, a local optimal solution is assumed to be reached. From local optimal solutions, the global solution is returned. Therefore, the solution of CLARANS is unbounded in terms of quality. By setting a higher value of maxneighbor, the quality of solution can be improved to approach the quality level of PAM and SHR. Inorder to compare the efficiency of SHR with CLARANS, suppose SHR and CLARANS have a common initial subset ω. In the best case of CLARANS, ω is the final solution such that no better point in D - ω can be found by conducting maxneighbor tentative replacements. Then, CLARANS returns ω. As suggested, maxneighbor is 1.25 percent of the number of the possible replacements between all points in D and all points in ω, and thus the number of subsets evaluated by CLARANS is 1. 25% k (|D|- k) in the best case. In SHR, each node at high level of R-tree tentatively replaces all p € ω and, following the best first traverse fashion, we visit the children of the node by replacing which the total distance can be reduced the most. The total number of subsets evaluated by SHR is m - n where m is the number of nodes visited and n is the fan-out of node. n is a constant in a given R-tree and the value is 50 in our experiments. In the best case of SHR, ω is the final solution such that m is $\log_n|D|$. For a medium size D (|D| = 20k in the PP data set used in our experiments),m*N in SHR is smaller than 1. 25% k (|D|- k)in CLARANS by several times in the best case. The larger the D, the better the relative performance of SHR, compared to CLARANS. In general cases, SHR is deliberately designed to always refine the currently most promising subset when traversing the index structure while CLARANS randomly selects a subset to process. Thus, we expect the performance of SHR to be generally better than CLARANS.

## 8 CONCLUSION

The GNG query can be found in various business analytic and decision support systems. Since GNG query is

a problem computationally intractable, both efficiency and effectiveness are critical to the practical value of GNG query. Use of hierarchical blocks instead of data points to optimize the number of subsets evaluated. This work proposes two GNG query processing algorithms, Exhaustive Hierarchical Combination algorithm and Subset Hierarchial Refinement. EHC provides the optimal solution when k = 2 and SHR provides fast approximate solution for any setting of k. Compared to PAM , SHR has the same quality while the performance is better by 1-3 orders of magnitude. Compared to CLARANS , SHR returns quality bounded solution and SHR outperforms CLARANS by several times in most cases. The performance of EHC, SHR and optimized SHR will be evaluated and compared.

**REFERENCES**

[1].Ke Deng, ShaziaSadiq, Xiaofang Zhou, Senior Member, IEEE, Hu Xu, Gabriel Pui Cheong Fung, and Yansheng Lu ,"On Group Nearest Group Query Processing" IEEE transactions on knowledge and data engineering, vol. 24, no.2, February2012

[2]. K. Deng, H. Xu, S. Sadiq, Y. Lu, G. Fung, and H. Shen, "Processing Group Nearest Group Query," Proc. 25th IEEE Int'l Conf. Data Eng., 2009.

[3]. K. Mouratidis, D. Papadias, and S. Papadimitriou, "Tree-Based Partition Querying: A Methodology for Computing Medoids in Large Spatial Datasets," The VLDB J., vol. 17, no. 4, pp. 923-945, 2008.

[4]. K. Deng, X. Zhou, and H. Shen, "Multi-Source Skyline Query Processing in Road Networks," Proc. 23th IEEE Int'l Conf. Data Eng., 2007.

[5]. Sharifzadeh and C. Shahabi, "The Spatial Skyline Queries,"Proc. 32nd Very Large Data Bases Conf., 2006. M. Yiu, N. Manoulis, and D. Papadias,"Aggregate Nearest Neighbor Queries in road Networks," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 6, pp. 820-833, June 2005.

[6]. D. Papadias, Y. Tao, K. Mourstidis, and C.K. Hui, "Aggregate Nearest Neighbor Queries in Spatial Databases," ACM Trans. Database Systems, vol. 30, no. 2, pp. 529-576, 2005.

[7]. D. Papadias, Q. Shen, Y. Tao, and K. Mouratids, "Group Nearest Neighbor Queries," Proc. 20th IEEE Int'l Conf. Data Eng., 2004.

[8] Local Search in Combinatorial Optimization, E. Aarts and J.K. Lenstra, eds. Princeton Univ. Press, 2003.

[9] J. Han, M. Kamber, and A. Tung, "Spatial Clustering Methods in Data Mining: A Survey," Geographic Data Mining and Knowledge Discovery, pp. 1-29, 2001.

**Author Profile**

K.Sheik Ibrahim received his B.Tech degree in Information Technology from Anna University, Chennai, India in 2011. He is currently pursuing his M.E in Computer and Communication engineering in Sethu Institute of Technology. His research interests include Wireless Networks and cryptography