COE: Clustering with Obstacles Entities A Preliminary Study

Anthony K. H. Tung Jean Hou Jiawei Han

School of Computing Science
Simon Fraser University
British Columbia
Canada V5A 1S6
Email: {khtung, jhou, han}@cs.sfu.ca

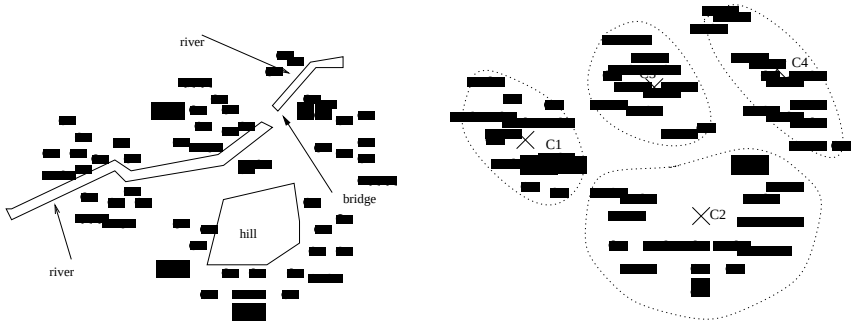
Abstract. Clustering analysis has been a very active area of research in the data mining community. However, most algorithms have ignored the fact that physical obstacles exist in the real world and could thus affect the result of clustering dramatically. In this paper, we will look at the problem of clustering in the presence of obstacles. We called this problem the COE (Clustering with Obstacles Entities) problem and provide an outline of an algorithm called COE-CLARANS to solve it.

1 Introduction

The studies of clustering on large databases started with the introduction of CLARANS [NH94] and since then, a tremendous amount of research had been made by the database community on this field [HK00].

Typically, a clustering task consists of separating a set of points into different groups according to some *measure of goodness* that differ according to application. For example, in market research, managers who are planning the location of their stores may wish to cluster their customers according to their location and then locate a store to serve each cluster. In such a case, a common measure of goodness will be the sum of square of the **direct** Euclidean distance between the customers and the centre of the cluster they belong to. However, in many real applications, the use of direct Euclidean distance has its weakness as illustrated by the following example.

Example 1. A bank manager wishes to locate 4 ATMs in the area shown in Figure 1a to serve the customers who are represented by points in the figure. In such a situation, however, natural obstacles exist in the area and they should not be ignored. This is because ignoring these obstacles will result in clusters like those in Figure 1b which are obviously wrong. Cluster C_1, for example, is in fact split by a river and some customers on one side of the river will have to travel a long way to the allocated ATM on the other side of the river.



(a) Customers' location and obstacles.

(b) Clusters formed when ignoring obstacles.

Fig. 1. Planning the location of ATMs

Example 1.1 illustrated a practical problem encountered by many users of traditional clustering algorithms: the lack of mechanism to integrate physical obstacles into clustering algorithms. In many application, the discovered clusters can be much more useful if they are found while keeping the physical imitation of the obstacles in mind.

Depending on the application on hand, different clustering algorithms will be needed and they will be affected differently by the existence of obstacle entities. In this paper, we will concentrate on adapting CLARANS to handle obstacles and we called the adapted algorithm **COE-CLARANS**. The problem in Example 1.1 is formally described as follows:

We are given a set P of n points $\{p_1, p_2, ..., p_n\}$ and a set O of m **non-intersecting** obstacles $\{o_1, ..., o_m\}$ in a two dimensional region, R with each obstacle o_i represented by a simple polygon. The distance, $d(p, q)$ between any two points, p and q, is defined as the length of the shortest Euclidean path from p to q without cutting through any obstacles. To distinguish this distance from the direct Euclidean distance, we will refer to this distance as *obstructed distance* in this paper. Our objective is to partition P into k clusters $C_1, ..., C_k$ such that the following square-error function, E, is minimized.

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} d^2(p, m_i)$$

where m_i is the centre of cluster C_i that is determined also by the clustering.

Due to lack of space, we will only outline the steps taken in COE-CLARANS to handle obstacles in the next section follow by the conclusion in Section 3.

2 The COE-CLARANS Algorithm

In order to adapt an existing clustering algorithm like CLARANS to handle obstacles, two different approaches can be adopted. The first is a loosely-coupled approach in which the obstacles are handled solely by the distance function and the clustering algorithm uses the distance function as a black box without catering for obstacles. The second approach is a tightly-coupled approach in which both the clustering algorithm and the distance function take obstacles into account. COE-CLARANS uses the second approach as it give more room for optimizing performance. COE-CLARANS use two techniques to perform efficient clustering. We will introduce them in this section.

2.1 Pre-clustering

To make COE-CLARANS efficient, a pre-clustering step similar to those in BIRCH [ZRL96], ScaleKM [BFR98] and CHAMELEON [KHK99] are taken to group the objects into a set of *clustering features* [ZRL96]. We call these clustering features, *micro-clusters*. There are two advantages in adding a pre-clustering step. First, the compressed micro-clusters take up much less memory space and clustering can thus be performed in main memory. Second, as computing the distance between objects and the cluster centers is an expensive operation, pre-clustering will help reduce the number of such operation.

In order to avoid having micro-clusters that are split by an obstacle, we first triangulate the region R into triangles and group the data points according to the triangle that they are in. Micro-clusters are then formed in each group separately. As points within a triangle are all mutually visible to each other, this ensures that micro-cluster formed are not split by an obstacle.

With the use of micro-clusters for clustering, we have to take note that the cluster centers are now micro-clusters and we are approximating the location of the actual medoids to be within these cluster centers.

2.2 Using the Lower Bound of E for Pruning

The CLARANS algorithm is a generate-and-test algorithm which randomly pick a cluster center o_i and try to replace it with a new center o_{random}. To judge whether o_{random} is a better center than o_i, the square error function E is computed with o_{random} as the cluster center and if it is found to be lower than the one computed with o_i as the center, replacement will take place. However, the computation of E is very expensive with the existence of obstacles. To avoid the unnecessary computation of E, an more easily computed **lower bound** of E, E' is first computed. If E' is already higher than the best solution so far, then o_{random} can be abandoned without the need for E to be computed.

To compute E' with o_{random} as a cluster center, we first underestimate the distance between o_{random} and the micro-clusters by using direct Euclidean distance. Thus, if the direct Euclidean distance between a micro-cluster p and

o_{random} is shorter than the obstructed distance between p and the other $k-1$ unchanged cluster centers, then p is assigned to o_{random} and the direct Euclidean distance between them will be used when computing the estimated square-error function E'. This makes E' a lower bound for the actual square-error function E. Since E' is a lower bound of E, we can choose to abandon o_{random} without computing E if E' is already higher than the square-error function of the best solution so far.

3 Conclusion

In this paper, we introduce the problem of COE which we believe is a very real and practical problem. We selected a clustering problem and outline an algorithm COE-CLARANS for solving it. COE-CLARANS makes use of two main ideas to enhance its efficiency. First, it uses the idea of pre-clustering to compress the dataset into micro-clusters which could be clustered in the main memory and thus avoids I/O overhead. Second, it avoids unnecessary computation by first estimating a lower bound E' for the square-error function E and then computes E only if E' proves to be lower than the best solution that has been found. We believe that there is still a lot of room for research in the problem of COE and hope that our work could motivate more people to look into this area.

References

[BFR98] P. Bradley, U. Fayyad, and C. Reina. Scaling clustering algorithms to large databases. In *Proc. 4th Int. Conf. Knowledge Discovery and Data Mining (KDD'98)*, pages 9–15, New York, NY, August 1998.

[CLR89] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithm.* The MIT Press, 1989.

[HK00] J. Han and M. Kamber. *Data Mining: Concepts and Techniques.* (to be published by) Morgan Kaufmann, 2000.

[KHK99] G. Karypis, E.-H. Han, and V. Kumar. CHAMELEON: A hierarchical clustering algorithm using dynamic modeling. *COMPUTER*, 32:68–75, 1999.

[NH94] R. Ng and J. Han. Efficient and effective clustering method for spatial data mining. In *Proc. 1994 Int. Conf. Very Large Data Bases*, pages 144–155, Santiago, Chile, September 1994.

[O'R98] J. O'Rourke. *Computational Geometry in C (2nd Edition).* Cambridge University Press, 1998.

[ZRL96] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: an efficient data clustering method for very large databases. In *Proc. 1996 ACM-SIGMOD Int. Conf. Management of Data*, pages 103–114, Montreal, Canada, June 1996.