ATLANTIS PRESS

Advances in Intelligent Systems Research, volume 133

2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE2016)

Balanced COD-CLARANS: A Constrained Clustering Algorithm to Optimize Logistics Distribution Network

Tong Zhang[1], Dong Wang[2] and Haonan Chen[2]

[1]Shanghai Jiao Tong University, China

[2]Shanghai Jiao Tong University, China

***Abstract*—In this paper, we focus on the problem of the siting of distribution stations, which is of key importance during designing a logistics distribution network. This problem includes dealing with physical obstacles in real world, making the orders as close as possible to their respective stations and making the workload of each station as balanced as possible. To solve this problem, we use a dataset of the geographical coordinates of all orders of a certain express company per day in Shanghai, and propose an algorithm called Balanced COD-CLARANS, which is a constrained clustering algorithm capable of handling physical obstacles and balance factor and outputting a set of clusters for decision-making. Besides, we design the experiment to prove that Balanced COD-CLARANS works well.**

Keywords-data mining; constrained clustering; logistics distribution network; balance-driven; obstacle

I. BACKGROUND AND PROBLEM STATEMENT

With e-commerce develops rapidly in recent years, more and more people have already accepted and get used to go shopping online. Meanwhile, people also expect more convenient shopping experience, such as receiving goods as early as possible and so on. All these requirements cannot be satisfied without logistics distribution network, so it has been one of the main bottlenecks preventing enterprises and shops from further expanding their markets. Therefore, the optimization of logistics distribution network is of great importance and necessity.

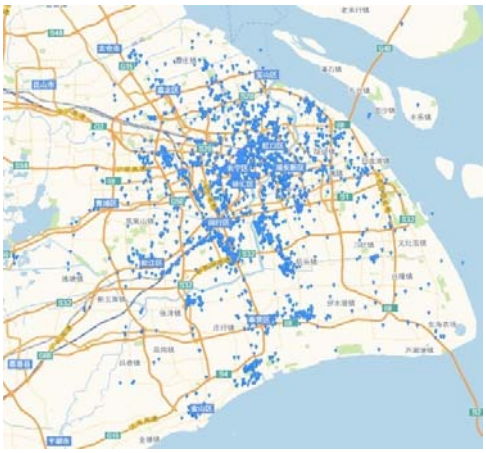

FIGURE I. THE DISTRIBUTION OF THE ORDERS IN SHANGHAI.

For the purpose of the optimization of logistics distribution network, we focus on the problem of the siting of distribution stations, which is of key importance during designing a logistics distribution network. At present, express companies often decide the initial locations of distribution stations according to experience. As the situation changes, a station may be divided into several ones for overload, or merge with another because of underload. This process has no theory support and lacks replicability. Thus, it may cause waste of labor, material and money.

Data mining is a hot topic in recent years. If an express company has a certain amount of geographical coordinates of orders and plans to change the stations' locations, or a online e-commerce platform having numerous geographical coordinates of orders wants to design its own logistics distribution network, both of them can mine useful "knowledge" through data mining to provide theories and references for the siting of distribution stations.

Based on this idea, we obtain a dataset of the geographical coordinates of all orders of a certain express company per day in Shanghai and use clustering to get a set of stations' locations. We take the obstacles in real world into account and aim at not only making the orders as close as possible to their respective stations but also making the workload of each station as balanced as possible.

The remainder of this paper is organized as follows. In Session 2, we summarize some related work. In Session 3, we introduce the algorithm called Balanced COD-CLARANS to optimize the logistics distribution network. Experimental results are analyzed in Section 4. Conclusions are given in Section 5.

II. RELATED WORK

Users often have background knowledge that they want to integrate into cluster analysis, and ignoring this knowledge may cause some unexpected results. Such information can be modeled as clustering constraints, and constrained clustering is proposed to take these constraints into account. The following are the constraints related to this paper: physical obstacles and balance.

There are many researches on physical obstacles. Tung et al. proposed COD-CLARANS[1], which first introduced obstacle

distance into CLARANS[2]. The concept of visible space is introduced into DBCluC[3] which is based on DBSCAN[4]. DBRS+[5] extending the density-based clustering method DBRS[6] can handle any combination of intersecting obstacles. AUTOCLUST+[7] using Delaunay diagram not only detects clusters of arbitrary shapes, but also clusters of different densities.

In balanced clustering, most researches focus on how to obtain an equal number of objects in each cluster. In general, it is a 2-objective optimization problem in which two aims contradict each other: to minimize MSE and to balance cluster sizes. In balance-constrained clustering, cluster size balance is a mandatory requirement. Constrained k-means[8], Balanced k-means[9], etc, belong to this category. On the other hand, balance is an aim but not mandatory in balance-driven clustering. FSCL[10], SRcut[11], etc, belong to this category.

In this paper, we don't focus on an equal number of objects but a balanced workload in each cluster. The workload of a cluster is a value computed more complicatedly, so the methods mentioned above no longer work. Besides, we take physical obstacles and balance into consideration at the same time. These are the challenges we meet.

III. BALANCED COD-CLARANS

A. *Algorithm Framework*

We choose the k-medoids method as our algorithm framework. The k-medoids method pick actual objects as 'medoids' to represent the clusters, using one representative object per cluster. This ensures that the medoids don't fall into some obstacle. Then it aims to partition n objects into k clusters in which each object belongs to the cluster with the nearest medoid, which results in a partitioning of the data space into Voronoi cells. This guarantees that the responsible regions of any pair of stations don't overlap. Moreover, there are two elements in the k-medoids method we can modify to solve our problem: the dissimilarity between two objects and the objective function which is used to judge the quality of the clusters. In general, we use Euclidean distance to measure the dissimilarity, and the objective function is defined as the sum of the dissimilarities between each object p and its corresponding representative object:

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

where E is the sum of the dissimilarities for all objects p in the data set, and o_i is the representative object of C_i. And the k-medoids method groups n objects into k clusters by minimizing this objective function.

Since the volume of our data sets after processed is nearly 10000. We choose CLARANS as our algorithm framework. It is based on PAM algorithm, and use random search to have lower algorithm complexity and deal with larger data sets than PAM.

Next, we modify the method used to measure the dissimilarity and the objective function in the general k-medoids method to deal with our problem.

B. *Obstacles*

There are many obstacles in real world, such as rivers, lakes and mountains, and if you do not consider these obstacles, the results will fail to meet your expectation. For example, the two objects on both sides of the river may be partitioned in the same cluster. Although the Euclidean distance between these two objects is very close, in fact the actual distance between them may be "far" because the bridge across the river is very far from them.

The solution to physical obstacles is basically the same as that proposed by Tung et al.[1]. In this solution, the length of the shortest path between two objects in the visibility graph, called obstacle distance, is proposed to replace Euclidean distance and is used to measure the dissimilarity.

A visibility graph is the graph, VG= (V,E), such that each obstacle vertex and each object in the data set has a corresponding node in V, and two nodes, v_1 and v_2, in V are joined by an edge in E if and only if the corresponding obstacle vertices or objects they represent are visible to each other. Please note that we use node to represent obstacle vertices and objects in this session. If two objects are visible to each other, the obstacle distance is equal to the Euclidean distance. Otherwise, the obstacle distance between them can consist of several Euclidean distances. Let us denote the set containing all the obstacle vertices visible to node x as vis (x), the Euclidean distance between node x and node y as d(x, y), and whether x and y are visible to each other as visible(x, y). Then the obstacle distance between node x and node y is defined recursively as:

$$d_{ob}(x,y) = \begin{cases} d(x,y) & if\ visible(x,y) = true \\ \min\limits_{u \in vis(x), v \in vis(y)} [d(x,u) + d_{ob}(u,v) + d(v,y)] & else \end{cases} \tag{2}$$

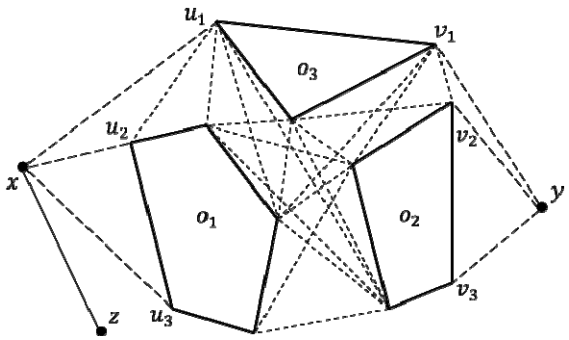

FIGURE II. THIS IS A VISIBILITY GRAPH. POLYGON O1, O2 AND O3 IN THIS GRAPH REPRESENT THREE OBSTACLES, AND X, Y AND Z REPRESENT THREE OBJECTS. SINCE X AND Z ARE VISIBLE TO EACH OTHER, THE OBSTACLE DISTANCE IS EQUAL TO THE EUCLIDEAN DISTANCE. MEANWHILE, X AND Y ARE NOT VISIBLE TO EACH OTHER. THEIR OBSTACLE DISTANCE CAN BE COMPOSED OF THE EUCLIDEAN DISTANCE BETWEEN X AND SOME OBSTACLE VERTEX U VISIBLE TO X, THE EUCLIDEAN DISTANCE BETWEEN Y AND SOME OBSTACLE VERTEX V VISIBLE TO Y, AND THE OBSTACLE DISTANCE BETWEEN U AND V.

To reduce the cost of distance computation between any two pairs of objects, we precompute the obstacle distance between any two pairs of obstacle vertex and vis(x) for every node x. Every time we compute the obstacle distance between two objects invisible to each other, we just enumerate all possibilities of (2), and take the minimum value as the result.

So how to compute vis(x) for a node x? The problem can be solved by a data structure called BSP tree[12], which can efficiently determine whether two node are visible to each other. No unnecessary details are given here.

Since the Euclidean distance is replaced by the obstacle distance, the possibility that two objects invisible to each other are partitioned into the same cluster effectively declines.

C. *Balanced Workload*

As mentioned in Session 1, we aim at not only making the orders as close as possible to their respective stations but also making the workload of each station as balanced as possible. Thus the clustering algorithm we proposed is balance-driven.

Since CLARANS partitions n objects into k clusters by minimizing the objective function, we modify the objective function and introduce the balance factor into it. The new objective function is defined as:

$$E' = \frac{\lambda}{k} * E + (1-\lambda)\sqrt{\frac{\sum_{c\in C}(W_c - \overline{W_c})^2}{k}}, \overline{W_c} = \frac{\sum_{c\in C} W_c}{k}, \lambda \in [0,1] \quad (3)$$

where E' is the linear combination of the mean of dissimilarities for all objects p in the data set and the standard deviation of workloads for all clusters, W_c is the workload for cluster c, C is the set of all clusters, of which size is k, and λ is the ratio to control the effect of balance on the result. CLARANS iteratively decrease the value of this objective function, and this means that in each iteration the objects are more close to their respective medoids and the workload of each cluster is more balanced. We also note that if we set λ to 1, the new objective function degenerates to the original one multiplied by 1/k and if we set λ to 0, the new objective only focuses on the balance factor. In (3), the workload of a cluster is a value. And this increases the flexibility of the algorithm, which means that we can refine the final result by improving the method of calculating the workload, and even we can use the new objective function in any balance-driven clustering problem where the factor required to be balanced can be measured by a value.

Since in real world the workload of delivering a lot of orders in a small region may be equal to the workload of delivering several orders in a large region, the workload of a cluster should be a value not only related to the number of objects in this cluster. Thus we define the workload of a cluster c as:

$$W_c = \sum_{p\in c} cost_p \quad (4)$$

where W_c is the sum of the costs for all objects p in the cluster c. The cost of an object is the delivery cost of the order which this object represents. In this paper, we simply define the cost of an object p as:

$$cost_p = d_{ob}(p, o_c) \quad (5)$$

where $cost_p$ is the obstacle distance between p and its medoid. In the future work, this can be improved to obtain a more appropriate result which is used to optimize the logistics distribution network.

D. *Conclusion*

So far, a new algorithm called Balanced COD-CLARANS to optimize the logistics distribution network is proposed. Besides the input dataset, this algorithm has four input parameters three of which are the same as those of CLARANS: the number of clusters, the maximum number of neighbors examined, the number of local minima obtained and the ratio to control the effect of balance on the result. Although our proposed Balanced COD-CLARANS is similar to CLARANS due to its framework based on CLARANS, there are three key points that are different from CLARANS:

- In the initialization phase Balanced COD-CLARANS applies the strategy of k-means++ so that it can avoid the initial centers getting together and reduce the number of iterations for convergence.
- Since Balanced COD-CLARANS introduces balanced workload in each cluster, the objective function is modified to guide the clusters to having balanced workload.
- Balanced COD-CLARANS calculates the obstacle distance between two objects instead the Euclidean distance in consideration of physical obstacles.

IV. EXPERIMENT

A. *Experimental Design*

There are three data sets used in the contrast experiment: the dataset of the geographical coordinates of all orders of a certain express company per day in Shanghai, the dataset of the physical obstacles in Shanghai, most of which are the Huangpu River and the waters of Shanghai, and the dataset of the locations and responsible regions of the existing stations decided by this express company. The first dataset is extracted from the order information table provided by the express company. The geographical coordinates of the obstacle vertices in the second dataset is obtained from the electronic map manually. The third dataset is crawled from the official website of the company website.

There are four groups set up for the contrast experiment. In group 1, all the orders are partitioned into the corresponding responsible regions of the existing stations. And each station and the orders it is responsible for are treated as one cluster. In group 2, we use CLARANS to partition the orders, without regard to physical obstacles and balanced workload. In group 3, CLARANS is used to partition the orders, without regard to

physical obstacles but balanced workload. In group 4, we use Balanced COD-CLARANS to partition the orders. Among all of the above groups, the input is the same, and the method of calculating the workload which use the obstacle distance is the same. Thus, the difference among the algorithms is the only factor that leads to different results.

B. Experimental Results

In Figure III to VI, green bars represent the workload of a cluster, and purple dots represent the number of the objects in a cluster. These figures show that in a cluster there is no direct correlation between the workload and the number of the objects, which is in accord with our thinking in Session 3. In this paper, since the workload of a cluster is defined as the sum of the obstacle distance between the medoid and every object p within the cluster, we can use it to measure the closeness of the objects to their respective medoids. In Figure VII, green bars represent the average of workloads for all the clusters, which means the closeness of the objects to their respective medoids. Purple bars represent the standard deviation of workloads for all clusters, which means the balance degree among the workloads of the clusters. The statistical results of the four groups are ordered by the sequence number of the group and arranged from left to right.

Figure III is the result of group 1. It shows that the workloads of most clusters are below the average, and there are some clusters with very high workload and ones with very low workload. Meanwhile, the average and the standard deviation of group 1 are the biggest among the four groups. This means that the workloads of the existing stations are not balanced to some extent, indicating that the siting of the stations is required to be optimized, and also showing that the work of this paper is valuable.

Figure IV is the result of group 2. We can see from the figure that the result has improved compared to the one of group 1, but the maximal workload of group 2 is higher than the one of group 1. The cause of this is that group 2 is without regard to physical obstacles. Compared to group 2, although the standard deviation doesn't decrease significantly, the average decreases by 36%, which optimizes the total workload of all the clusters.

Figure V is the result of group 3. It shows that the result is more balanced compared to the one of group 2, but there are a few clusters with workload 5 times higher than the average. The cause of this is also that group 3 is without regard to physical obstacles. Besides, we can see that the average of group 3 is higher than the one of group2, because group 2 takes balance into account and sacrifices some closeness of the objects to their respective medoids.

Figure VI is the result of group 4. The workload distribution of all the clusters in this group is the most balanced one among all the groups, which can be from the fact that the maximal workload is the smallest one in the four groups. Meanwhile, we can see that the maximum value of the Y-axis in figure VI is only one-tenth of the ones in figure III, IV and V. Besides, the average and the standard deviation of group 4 are the smallest among the four groups. Especially, the standard deviation is 14% of the one of group1, 16% of the one of group2, and 22% of the one of group3, which means that Balanced COD-CLARANS works well.

V. CONCLUSIONS

In this paper, we study the optimization problem of the siting of distribution stations. To solve this problem, we propose algorithm called Balanced COD-CLARANS, which is a constrained clustering algorithm capable of handling physical obstacles and balance factor and outputting a set of clusters for decision-making. And we design the experiment to prove that Balanced COD-CLARANS works well.

REFERENCES

[1] Tung, A. K., Hou, J., & Han, J. (2001). Spatial clustering in the presence of obstacles. In Data Engineering, 2001. Proceedings. 17th International Conference on (pp. 359-367). IEEE.

[2] Ng, R. T., & Han, J. (2002). CLARANS: A method for clustering objects for spatial data mining. IEEE transactions on knowledge and data engineering, 14(5), 1003-1016.

[3] Zaïane, O. R., & Lee, C. H. (2002). Clustering spatial data when facing physical constraints. In Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on (pp. 737-740). IEEE.

[4] Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In Kdd (Vol. 96, No. 34, pp. 226-231).

[5] Wang, X., Rostoker, C., & Hamilton, H. J. (2004, September). Density-based spatial clustering in the presence of obstacles and facilitators. In European Conference on Principles of Data Mining and Knowledge Discovery (pp. 446-458). Springer Berlin Heidelberg.

[6] Wang, X., & Hamilton, H. J. (2003, April). DBRS: a density-based spatial clustering method with random sampling. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 563-575). Springer Berlin Heidelberg.

[7] Estivill-Castro, V., & Lee, I. (2001). Autoclust+: Automatic clustering of point-data sets in the presence of obstacles. In Temporal, Spatial, and Spatio-Temporal Data Mining (pp. 133-146). Springer Berlin Heidelberg.

[8] Bradley, P. S., Bennett, K. P., & Demiriz, A. (2000). Constrained k-means clustering. Microsoft Research, Redmond, 1-8.

[9] Malinen, M. I., & Fränti, P. (2014, August). Balanced k-means for clustering. In Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR) (pp. 32-41). Springer Berlin Heidelberg.

[10] Banerjee, A., & Ghosh, J. (2004). Frequency-sensitive competitive learning for scalable balanced clustering on high-dimensional hyperspheres. IEEE Transactions on Neural Networks, 15(3), 702-719.

[11] Burkhard, R., Dell'Amico, M., & Martello, S. (2012). Assignment Problems (Revised reprint).

[12] Fuchs, H., Kedem, Z. M., & Naylor, B. F. (1980, July). On visible surface generation by a priori tree structures. In ACM Siggraph Computer Graphics(Vol. 14, No. 3, pp. 124-133). ACM.

FIGURE III. THIS IS THE WORKLOAD DISTRIBUTION OF GROUP 1.

FIGURE IV. THIS IS THE WORKLOAD DISTRIBUTION OF GROUP 2.

FIGURE V. THIS IS THE WORKLOAD DISTRIBUTION OF GROUP 3.

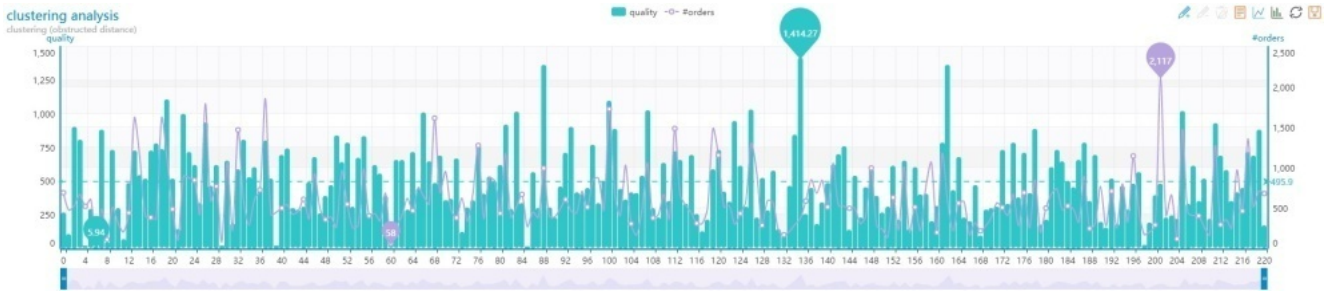

FIGURE VI. THIS IS THE WORKLOAD DISTRIBUTION OF GROUP 4.

FIGURE VII. THIS IS THE AVERAGE AND THE STANDARD DEVIATION OF THE WORKLOAD FOR THE FOUR GROUPS.