

On the Techniques for Data Clustering with Numerical Constraints

Bi-Ru Dai, Cheng-Ru Lin, and Ming-Syan Chen ^{*†}

Abstract

In this paper, the attributes employed to model the constraints are called constraint attributes and those attributes involved in the objective function to be optimized are called cost-optimal attributes. The constrained clustering considered is conducted in such a way that the objective function of cost-optimal attributes is optimized subject to the condition that the imposed constraint is satisfied. Explicitly, we address the problem of constrained clustering with numerical constraints, in which the constraint attribute values of any two data items in the same cluster are required to be within the corresponding constraint range. We devise an effective and efficient algorithm with complete-link to solve this clustering problem. It is noted that due to the intrinsic nature of the numerical constrained clustering, there is an order dependency on the process of attaining the clustering, which in many cases degrades the clustering results. In view of this, we devise a *progressive constraint relaxation* technique to remedy this drawback and improve the overall performance of clustering results. Explicitly, by using a smaller (tighter) constraint range in earlier iterations of merge, we will have more room to relax the constraint and seek for better solutions in subsequent iterations. It is empirically shown that the progressive constraint relaxation technique is able to improve not only the execution efficiency but also the clustering quality.

Keywords: Data Mining, data clustering, constrained clustering

1 Introduction

Data clustering is a useful technique for many applications, including similarity search, pattern recognition, trend analysis, marketing analysis, grouping, classification of documents, and so forth [3][7][10]. In data clustering, similar data points are grouped together in a cluster. Since the early work in k-means algorithm, the data clustering has been studied for many years and several technologies have been developed, including

the nearest neighbor clustering [12], fuzzy clustering [1], partitional clustering [4], hierarchical clustering [14], artificial neural networks for clustering [8], and so on. In addition, several works have been conducted for studying the constrained data clustering problems. The work in [15] defines a taxonomy of constraints for clustering with the focus on exploring the constraints which can be formulated with SQL aggregates. Some works are proposed to cope with the total mass constraints [2][13]. Also, the work in [11] focuses on the continuous constraint. The works for segmenting a video into story units try to collapse visually similar and temporally local shots into a compact structure by introducing the time-constraint into clustering problems [17]. The clustering techniques for spatial data in presence of physical constraints are discussed in [5][16][18].

Since data mining is an application dependent technology, the information involving domain knowledge is usually imposed on the mining systems as various constraints. A specific constrained clustering model is introduced in this paper to cope with these user specified constraints. Specifically, those attributes employed to model the constraints are called constraint attributes whereas those attributes involved in the objective function to be optimized, similar to those in most prior works, are called cost-optimal attributes. Note that an attribute could be a constraint attribute and a cost-optimal attribute at the same time. The constrained clustering considered in this paper is conducted in such a way that the objective function of cost-optimal attributes is optimized subject to the condition that the imposed constraint is satisfied. Explicitly, we address in this paper the problem of constrained clustering with numerical constraints, in which the constraint attribute values of any two data items in the same cluster are required to be within the corresponding constraint range. Note that such a clustering with numerical constraints is called for in many real applications. For example, we may apply this numerical constrained clustering on the basic data of people in a club to group people provided that we require the range of ages in each group to be no more than 5 years. Also, in order to observe the trends of any data sets with a timestamp, such as Web logs, video images, or CDR (i.e. Call Detail Record) of a moving person, a time constraint range is expected to

^{*}Electrical Engineering Department, National Taiwan University, Taipei, Taiwan, ROC

[†]Email: mschen@cc.ee.ntu.edu.tw, {owenlin, brdai}@arbor.ee.ntu.edu.tw

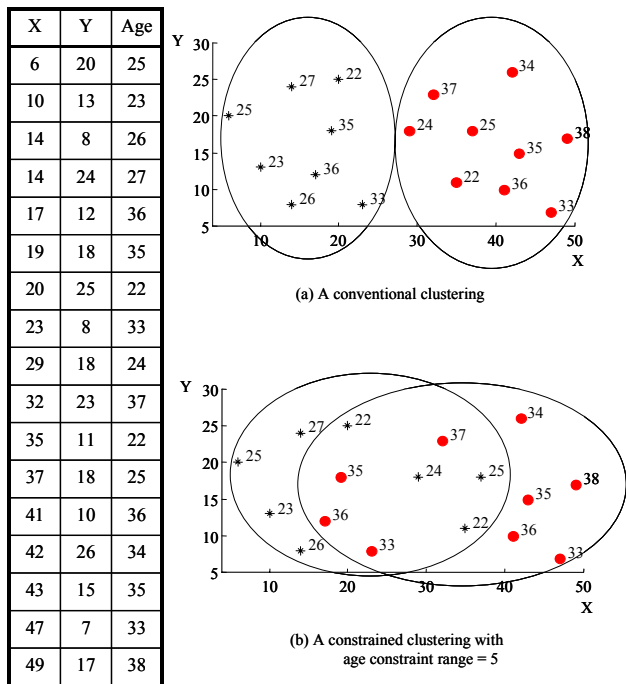


Figure 1: The difference between a conventional clustering and a numerical constrained clustering.

be specified to monitor the behavior of each data group, e.g., 5 minutes for video images or 24 hours for a Web log or a CDR. Such problems do require the clustering with numerical constraints. This numerical constrained clustering problem can be best understood by the following example. Consider the data points in Figure 1 where X and Y are the conventional cost-optimal attributes which form the coordinates of the residential location, and age is the constraint attribute with the constraint range being 5 years. The nodes in Figure 1(a) are identical to those in Figure 1(b) and the number next to each node represents the age of that person. By conventional clustering methods which are designed to cluster nearby nodes together, these nodes may be partitioned into the two clusters as shown in Figure 1(a), in which, however, the age constraint is not obeyed. Instead, one possible solution to this constrained clustering problem is shown in Figure 1(b), where members within 22 – 27 years old are in one cluster, and people within 33 – 38 years old are in the other cluster. Note that a conventional clustering method cannot resolve the constraint directly. Consequently, new methods are called for to handle this constrained clustering problem.

In view of this, we devise a hierarchical algorithm, namely the *constrained clustering with complete-link* (abbreviated as CCL) to handle the constraints by

merging valid clusters one by one. It is noted that due to the intrinsic nature of the numerical constrained clustering, there is an order dependency on the process of attaining the clustering, which in many cases degrades the clustering results. In view of this, we devise a *progressive constraint relaxation* technique to remedy this drawback and improve the overall performance of clustering results.

The rest of this paper is organized as follows. The problem description is given in Section 2. Section 3 presents the proposed algorithm to deal with this constrained clustering problem. The issue of order dependency is identified and solved by the progressive relaxation technique in Section 4. We empirically evaluate the performance of proposed algorithm in Section 5. This paper concludes with Section 6.

2 Problem Description

As mentioned earlier, a specific constrained clustering model is introduced in this paper to cope with the user specified constraints. Among all attributes of the data set, some are specified as *constraint attributes* and some are *cost-optimal attributes*. Note that an attribute could be a constraint attribute and a cost-optimal attribute at the same time because some attributes are possibly important in both considerations. Similar to the conventional clustering problems, there is an objective function operating on the cost-optimal attributes to measure the cost of clustering. In addition, an additional *constraint range* R_{a_C} is set for each constraint attribute a_C . For any pair of objects o_i, o_j in a cluster, the constraint distance $d_{a_C}(o_i, o_j)$ is required to be less than or equal to R_{a_C} , where $d_{a_C}(o_i, o_j)$ is the distance of constraint attribute a_C between any two objects o_i and o_j . Formally, we have the following problem definition.

Clustering with Numerical Constraints : Given a data set D of n objects $\{o_1, o_2, \dots, o_n\}$, and a predetermined number of clusters k , the numerical constrained clustering problem is defined as the problem of determining the k -clustering $Cl = \{C_1, C_2, \dots, C_k\}$ in such a way that the total cost $Cost(Cl)$ is minimized subject to the condition that for any pair of objects (o_i, o_j) in a cluster, $d_{a_C}(o_i, o_j) \leq R_{a_C}$, where a_C is any constraint attribute and R_{a_C} is the constraint range of a_C . The cost function $Cost(Cl)$ is calculated based on the cost-optimal attributes.

Without loss of generality, the number of constraint attributes is assumed to be one for ease of exposition, and the algorithms can be easily extended to multiple constraints. Note that as the constraint range approximates to infinity, this clustering problem can be reduced to a traditional clustering problem which has been proved as NP hard in [6]. Hence, we have the

following theorem for the complexity of the numerical constrained clustering problem.

THEOREM 2.1. *The problem of finding the optimal numerical constrained clustering is NP hard.*

Theorem 2.1 justifies our efforts in the following sections to explore efficient heuristics to solve this numerical constrained clustering problem.

3 Constrained Clustering Algorithm

In hierarchical algorithms, each data point initially forms a cluster by itself and then the algorithm repetitively merges the nearest clusters together until there are k clusters left over. We present algorithm CCL which is devised by utilizing the complete-link [9] clustering method in this section. The complete-link algorithm uses the distance of two farthest points as the inter-cluster distance. The distance measurement of two clusters is modified for this constrained clustering problem:

$$\widehat{dist}(C_i, C_j) = \begin{cases} dist(C_i, C_j), & \text{if } d_{a_C}(C_i, C_j) \leq R_{a_C}, \\ \infty, & \text{otherwise,} \end{cases}$$

where $dist(C_i, C_j)$ is the inter-cluster distance and $d_{a_C}(C_i, C_j)$ is the constraint distance between two clusters C_i and C_j . Hence, with these provisions, the complete-link clustering algorithm is revised to deal with the numerical constraints.

Algorithm CCL: Constrained Clustering with Complete-Link

//Input: an input data set, the number of clusters, k , and the constraint range R_{a_C}

1. Initially, each data point forms a cluster by itself.
2. The algorithm repetitively merges the two closest clusters.
3. Repeat Step 2 until exactly k clusters left or all pairs of points exceed the constraint range.

As the algorithm terminates, if the resulting cluster number k' is larger than k , those points in the smallest $(k' - k)$ clusters that contain the minimal number of points will be regarded as outliers so that exactly k large clusters are obtained. Note that algorithm CCL can also be modified to a constrained clustering method based on single-link algorithm by changing the inter-cluster distance measurement [14]. However, this method based on single-link is found to incur a much larger clustering cost and will thus not be explored in the following discussion.

4 Progressive Constraint Relaxation

It is noted that the hierarchical clustering algorithms always search for the nearest pair of clusters and merge them into a new single cluster. However, the merging process in an early iteration will reduce the number of possible merges in the subsequent iterations. This problem is referred to as order dependency in this paper. Note that the order dependency is different from the well-known chaining-effect in the single-link clustering algorithm, which means that two different clusters may be merged by a very slim noise link. To remedy the effect of order dependency, we devise a technique, called *Progressive Constraint Relaxation*, to take the future mergings into consideration by progressively relaxing the constraints in early iterations.

The basic idea of the progressive constraint relaxation is that by using a tighter (smaller) constraint range in early iterations of merging, we will have more room to seek for better solutions in subsequent iterations. In addition to the relaxation of the constraint range, the desired cluster number should also be temporarily modified accordingly. The whole relaxation process starts with a small local constraint range and a large local desired cluster number. The procedure continuously relaxes the constraint range and reduces the number of desired clusters, and eventually results in the actual constraint range and the cluster number specified by the user. Explicitly, a new parameter *level* is imported to control the number of relaxation steps. A constrained clustering algorithm is executed *level* times to achieve the final clustering result. If the value of *level* is equal to one, the algorithm is executed only once with the real constraint range and the cluster number. Otherwise, the temporary constraint range and cluster number, named *local_* R_{a_C} and *local_* k , are assigned by the following expressions,

$$local_k = \frac{n \times (level - i) + k \times i}{level},$$

$$local_R_{a_C} = R_{a_C} \times \frac{k}{local_k},$$

where n is the size of the data set, i means the current iteration number (from 1 to *level*), and k is the desired cluster number. Note that by this definition, the product of *local_* R_{a_C} and *local_* k is a constant, i.e., $local_R_{a_C} \times local_k = R_{a_C} \times k$.

As will be shown in our experiments, this technique improves not only the clustering quality but also the execution efficiency of the hierarchical algorithm.

Progressive Constraint Relaxation Technique

//Input: an input data set, the number of clusters, k , the relaxation level, and the constraint range R_{a_C}

1. For $i = 1$ to $level$, do Step 2 and Step 3.
2. Calculate the $local_k$ and the $local_R_{ac}$.
3. Run the constrained clustering algorithm based on $local_k$ and $local_R_{ac}$.

Algorithm CCL enhanced by the progressive constraint relaxation technique is referred to a progressive CCL. We then have the following theorem.

THEOREM 4.1. *With the progressive constraint relaxation technique, the time complexity of algorithm CCL becomes $O(nr_ck \log(\frac{nr_ck}{level})) + O(\frac{n^2 r_c}{level^2} \log(\frac{n^2 r_c}{level^2})) + O(f \times level)$, where $O(f)$ represents the lower order terms which are ignored in the analysis of CCL.*

Proof. To make use of the progressive constraint relaxation technique on the above algorithm, the time complexity of each iteration should be summed up. For the algorithm progressive CCL, the complexity is $\sum_{i=1}^{level} [O(n_i^2 r_{ci} \log(n_i^2 r_{ci})) + O(f)]$, where n_i is the number of remaining points at the beginning of iteration i , and r_{ci} is the constraint ratio of this iteration. Note that n_i is equal to the $local_k$ of its previous iteration,

$$\begin{aligned} & \sum_{i=1}^{level} [O(n_i^2 r_{ci} \log(n_i^2 r_{ci})) + O(f)] \\ = & \sum_{i=1}^{level} [O((local_k_{i-1}^2 \times r_c \frac{k}{local_k_i}) \\ & \times \log(local_k_{i-1}^2 \times r_c \frac{k}{local_k_i})) + O(f)], \end{aligned}$$

where

$$\begin{aligned} & O(local_k_{i-1}^2 \times r_c \frac{k}{local_k_i}) \\ = & O(\frac{(n(level - (i - 1)) + k(i - 1))^2}{level})^2 r_c \frac{k}{\frac{n(level - i) + ki}{level}}) \\ = & O(\frac{r_c k}{level} \times \frac{(n(level - i) + ki + (n - k))^2}{n(level - i) + ki}). \end{aligned}$$

For $i = 1$ to $level - 1$,

$$O(\frac{r_c k}{level} \times \frac{(n(level - i) + ki + (n - k))^2}{n(level - i) + ki}) = O(\frac{nr_ck}{level}).$$

For $i = level$,

$$O(\frac{r_c k}{level} \times \frac{(n(level - i) + ki + (n - k))^2}{n(level - i) + ki}) = O(\frac{n^2 r_c}{level^2}).$$

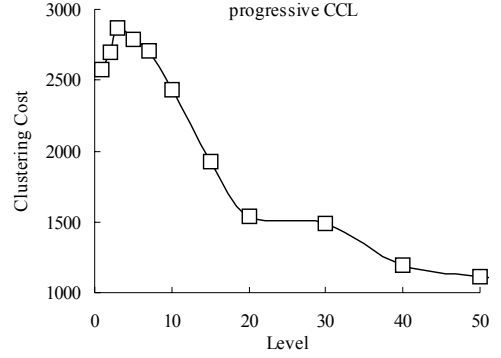


Figure 2: The clustering costs of progressive CCL at different levels.

Therefore, the time complexity is

$$\begin{aligned} & \sum_{i=1}^{level} [O(n_i^2 r_{ci} \log(n_i^2 r_{ci})) + O(f)] \\ = & \left(\sum_{i=1}^{level-1} O\left(\frac{nr_ck}{level} \log\left(\frac{nr_ck}{level}\right)\right) \right) \\ & + O\left(\frac{n^2 r_c}{level^2} \log\left(\frac{n^2 r_c}{level^2}\right)\right) + O(f \times level) \\ = & O\left(nr_ck \log\left(\frac{nr_ck}{level}\right)\right) + O\left(\frac{n^2 r_c}{level^2} \log\left(\frac{n^2 r_c}{level^2}\right)\right) \\ & + O(f \times level). \end{aligned}$$

Q.E.D.

The progressive constraint relaxation technique significantly reduces the time complexity of the constrained clustering algorithm. Note that utilizing the progressive constraint relaxation technique will not increase the space complexity.

5 Performance Studies

To assess the performance of algorithm progressive CCL, we have conducted an experiment on a computer with a 800Mhz Intel CPU and 1GB of memory. We will show the effect of progressive constraint relaxation technique on improving both the clustering quality and the execution efficiency of algorithm CCL. In the following experiments, we use the average squared error as the evaluation function for clustering results.

In this experiment, we apply progressive CCL on a data set of 5000 points with their timestamp in the range of (0, 10000). Here, we set the time constraint as 4000 and try to partition the data set into 20 clusters. Note that if the value of $level$ is equal to one, the original algorithm without relaxation is performed. As

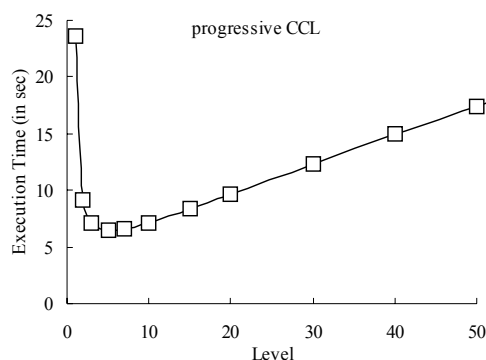


Figure 3: The execution times of the progressive CCL at different levels.

shown in Figure 2, by solving the order dependency, the progressive constraint relaxation technique improves the clustering results of algorithm CCL. As shown in Figure 3, the progressive constraint relaxation technique enhancement can also be used to reduce the execution time with smaller levels. The improvement could be as much as 75%, i.e., taking only $\frac{1}{4}$ of the original execution time.

6 Conclusions

In this paper, we proposed a new constrained clustering problem, named numerical constrained clustering. Explicitly, we addressed in this paper the constrained clustering with numerical constraints and devised an effective and efficient algorithm to solve it. In addition, we devised a *progressive constraint relaxation* technique to handle the order dependency and improve the overall performance of clustering results. As shown in the complexity analyses and also validated by our empirical studies, algorithm progressive CCL is executed very efficiently.

Acknowledgement

The authors are supported in part by the Ministry of Education Project No. 89E-FA06-2-4-7 and the National Science Council, Project No. NSC 91-2213-E-002-034 and NSC 91-2213-E-002-045, Taiwan, Republic of China.

References

[1] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, NY, 1981.

[2] P. S. Bradley, K. P. Bennett, and A. Demiriz. *Constrained K-Means Clustering*. MSR-TR-2000-65, Microsoft Research, May 2000.

[3] M.-S. Chen, J. Han, and P. S. Yu. Data mining: An overview from database perspective. *IEEE Trans. On Knowledge And Data Engineering*, 5(1):866–883, Dec. 1996.

[4] R. C. Dubes. How many clusters are best? - an experiment. *Pattern Recognition*, 20(6):645–663, 1987.

[5] V. Estivill-Castro and I. Lee. Autoclust+: automatic clustering of point-data sets in the presence of obstacles. In *International Workshop on Temporal, Spatial and Spatio-Temporal Data Mining (TSDM2000)*, pages 133–146, 2000.

[6] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: Freeman and Company, 1979.

[7] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000.

[8] J. Hertz, A. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation*. Addison-Wesley Longman Publ. Co., Inc., Reading, MA., 1991.

[9] B. King. Step-wise clustering procedures. *J. Am. Stat. Assoc.*, 69:86–101, 1967.

[10] C.-R. Lin and M.-S. Chen. A robust and efficient clustering algorithm based on cohesion self-merging. In *Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002)*, July 2002.

[11] C.-R. Lin and M.-S. Chen. On the optimal clustering of sequential data. In *Proceedings of the 2nd SIAM International Conference on Data Mining*, April 2002.

[12] S. Y. Lu and K. S. Fu. A sentence-to-sentence clustering procedure for pattern analysis. *IEEE Trans. Syst. Man Cybern*, 8:381–389, 1978.

[13] K. Rose, E. Gurewitz, and G. Fox. Constrained clustering as an optimization method. *IEEE Transactions On Pattern Analysis and Machine Intelligence*, 15(8):785–794, August 1993.

[14] P. H. A. Sneath and R. R. Sokal. *Numerical Taxonomy*. Freeman, London, UK, 1973.

[15] A. K. H. Tung, J. Han, L. V. S. Lakshmanan, and R. T. Ng. Constraint-based clustering in large databases. In *Proceedings of 2001 International Conference on Database Theory*, Jan. 2001.

[16] A. K. H. Tung, J. Hou, and J. Han. Spatial clustering in the presence of obstacles. In *International Conference On Data Engineering (ICDE)*, pages 359–367, 2001.

[17] M. Yeung and B. Yeo. Time-constrained clustering for segmentation of video into story units. In *International Conference on Pattern Recognition*, pages 375–380, May 1996.

[18] O. R. Zaiane, A. Foss, C.-H. Lee, and W. Wang. On data clustering analysis: Scalability, constraints, and validation. In *PAKDD2002*, pages 28–39, 2002.