禁忌搜尋之群聚分析
Cluster analysis based on the tabu search*

林聰武
Tsong-Wuu Lin

東吳大學資訊科學系
Dept. of Information and Computer Science
Soochow University

李宏儒
Horng-Ru Lee

輔仁大學數學研究所
Institute of Mathematics
Fu Jen Catholic University

摘要

群聚運算將m個物件分配到g個群中，同一個群中的成員具有類似的特徵，將物件的特徵以n度空間的點座標來表示，因此點與該群中心的距離可以表達物件在該群的適當性，群聚運算的目標在於將所有點至其群中心距離的總和最小化。這個問題有許多的局部最小值，一個植基於禁忌搜尋的增強演算法被用來解決該問題，一些產生嚐試解的策略被提出來增進

關鍵詞：禁忌搜尋、群聚分析、演算法、搜尋

Abstract

The clustering operation is to classify *m* objects into *g* groups whose members are similar in the interesting features. The features of an object are represented as a point within an n-dimensional Euclidean space. The distance between a point and the center of its group indicates fitness of the object within the group. The objective is to divide these objects into groups such that the total sum of these distances is minimized. There are many local minima in the problem. An improved algorithm is proposed to solve the problem based on the tabu search technique. Some strategies for generating trial solutions are presented to improve its effectiveness. From the experiments, our strategies are very powerful than existing methods.

Index: tabu search, cluster analysis, algorithm, search

1. Introduction

Cluster analysis may be defined as the process of separating a set of objects into groups, whose members are similar as much as possible, according to predefined criteria. It plays an essential role in fields of image processing, pattern recognition, medical research, artificial intelligence, geosciences, behavioral and social sciences, etc.

For the simplicity, objects are represented by points for cluster analysis in *n*-dimensional Euclidean space. The features of an object are kept as values of elements in its coordinate. The similarity of two objects can be measured by the distance between these corresponding points. The distance between a point and the center of its group indicates the fitness of the point (object) within the group. Smaller distance indicates better fitness. Therefore cluster analysis is changed into classifying *m* given points into *g* groups such that the total sum of distances between points and centers of their groups is minimized.

The total sum of these distances is the objective function of cluster analysis. It is non-convex and hence the analysis may suffer local minimum solutions which are not necessarily optimal[9]. The optimal solution is theoretically possible to get by examining all possible clusterings. The number of all possible clusterings for classifying *m* points into *g* groups is $S(m, g)$[3,12], where

$$S(m,g)=\frac{1}{g!}\sum_{i=0}^{g}(-1)^{i}\binom{g}{g-i}(g-i)^{m}$$

It is a very large number. For example, the number of enumerations is about 7.4 × 10^{32} for classifying 50 data points into 5 groups[11]. The exhaustive enumeration is not feasible in practice due to limitations of computer storage and time.

Thus, approximate heuristic techniques seeking a compromise or looking for a local minimum solution which is not necessarily global have usually been adopted. In this paper, we propose an efficient algorithm based on a tabu search technique for the clustering problem. Generating a trial solution from the current solution is a critical operation for the tabu search technique. If there are good guidelines for generating a trial solution that is closer to the optimal solution than the current solution, then the speed of searching a feasible solution is improved. Many strategies are proposed to generate trial solutions. From the

* A part of this work is supported by the National Science Council, Taiwan, republic of China, under Grant NSC87-2213-E-031-002.

experiments, these strategies are superior to one that is based on a probability threshold[2].

2. Cluster analysis

The supervised clustering problem may be stated as follows. Given m points in R^n, assign each point to one of g groups such that the sum of distances between each point and the center of its cluster is minimized. Therefore supervised clustering problem can be mathematically described as follows.

$$\text{Minimize } D(w, c) = \sum_{i=1}^{m} \sum_{j=1}^{g} w_{ij}\, d(x_i, c_j)$$

Subject to

$$\begin{cases} \sum_{j=1}^{c} w_{ij} = 1 \text{ for } i = 1, 2, \ldots, m \\ w_{ij} = 0 \text{ or } 1 \text{ for } i = 1, \ldots, m \text{ and} \\ \qquad j = 1, \ldots, g \end{cases}$$

where

c_j: center of the jth group (to be determined)

$d(x_i, c_j)$: distance between ith point and the center of jth group (to be found)

g: number of groups (given)

m: number of points (given)

w_{ij}: association weight of point x_i with group j (to be found)

x_i: location of the ith given point (given)

Many distance functions, such as Euclidean distance, city block distance, and chessboard distance, have been used for many applications in the real world. While clustering is applied to different applications, different distance functions are used to suit the needs of the application. The distance between point x_i and center of its group c_j can be calculated directly when the distance function is specified. In this paper, the Euclidean distance is used to measure similarity between two points.

From the definition above, there are two main problems to be solved in the supervised clustering. They are center determination and group allocation. To assign a point as the center of one group is called the center determination. The center of a group may be a given point or a virtual point (which is not one of given points). A given point is specified to a group by the group allocation process. A point is assigned to a group if the distance between it and the center of the group is less than the ones between it and centers of other groups.

The center determination and the group allocation are interrelated. Points can be assigned to groups when centers of groups are determined. After group allocation is completed, centers of groups are reconstructed by the following equation.

$$c_j = \frac{\sum_{i=1}^{m} w_{ij} x_{ij}}{\sum_{i=1}^{m} w_{ij}}$$

When centers of groups are changed, group allocation process is executed again. A stable situation will be obtained by repeating these processes above. Unfortunately, the obtained stable situation may be one local minima. It is not an absolute minima.

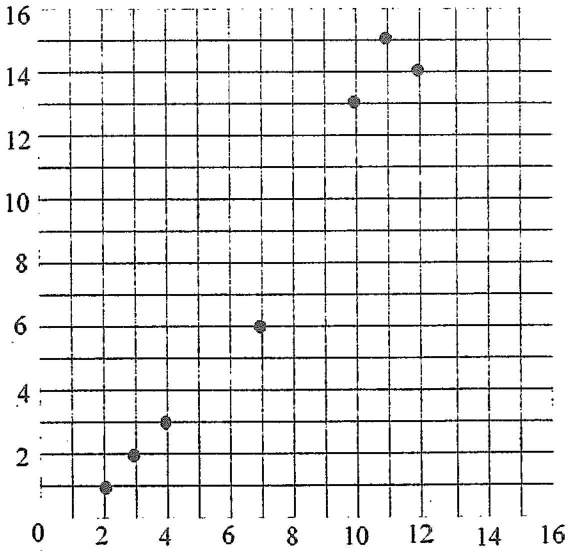

Figure 1: An example

An example is given in Figure 1 to demonstrate that a stable situation is a local minima. The clustering problem is to classify the following seven 2D points, (2, 1), (3, 2), (4, 3), (7, 6), (10, 13), (11, 15), and (12, 14), into 3 groups. If points are assigned into 3 groups as {(2, 1), (3, 2), (4, 3), (7, 6)}, {(10, 13)}, and {(11, 15), (12, 14)}, then a local minima is obtained. Its global minima is occurred when 3 groups are {(2, 1), (3, 2), (4, 3)}, {(7, 6)}, and {(10, 13), (11, 15), (12, 14)}.

A brute-force approach for the clustering problem is to assign points into one group randomly. Then the objective function value of each assignment is computed. The assignment whose objective function value is closest to optimal is chosen. However, testing different assignments is considered impracticable, especially for a large number of clusters[8]. For example, the number of enumerations of every possible partition of 56 data points into three clusters is greater than 10^{18} for the crude-oil data[10]. Many approaches are proposed for the clustering problem. A brief summary of these works is as follows.

Clustering techniques may be divided into two categories: hierarchical and partitional strategies.

Hierarchical clustering refers to a clustering process that organizes the data into large groups, which contain smaller groups, and so on. A hierarchical clustering can be drawn as a tree. The finest grouping is at the bottom of the tree, where each point by itself forms a group. The coarsest grouping is at the top of the tree, where exists a single group containing all points. In between, there are various numbers of clusters. At a level, two groups are merged into a group if distance between these two groups is the shortest among distances of pairwise groups. Different hierarchical clustering algorithms are obtained by using different methods to measure the distance between two groups.

Among the partitional clustering techniques, the K-means, C-means, or Isodata algorithm is the most widely used method. The K-means algorithm[4, 6] is based on the optimization of a specified objective function. A generalized convergence theorem for the method is derived[9]. In the theorem, conditions for local optimality of the solution obtained by the method are characterized. It has been proved that K-means algorithm terminates[9]; that is, eventually no points change groups. However, it may converge to a local minimum solution.

The isodata algorithm is developed to solve the clustering problem[5]. it can be considered to be an enhancement of the approach taken by the K-menas algorithm. The number of groups in it may be an interval. A group is split if the number of groups is too few or if the group contains very dis-similar points. Groups are merged if the number of groups grows too large or if groups are too close together. It is more complex and more expensive to implement than the k-means algorithm. But it suffers from local minimum solutions.

The genetic algorithm has applied to solve the clustering problem. Genetic algorithms are stochastic search methods based on the principle of natural genetic systems[7]. Each solution is usually coded as a binary string of finite length. Each string is considered as an individual. A collection of individuals is called a population. The best string obtained so far is preserved in a separate location outside the population so that the algorithm may report the best value found, among all possible solutions inspected during the whole process. The proposed method based on genetic algorithms can find lower value of the objective function than K-means.

A tabu search technique is used to solve the clustering problem[2]. It is different from the well-known hill-climbing local search techniques in the sense that it does not become trapped in local optimal solutions. The tabu search approach allows movements out of a current solution. Such movements may make the objective function worse but these movements eventually might achieve a better solution. Its results are better than the well-known K-means algorithm.

3. Cluster analysis using tabu search

Our improved algorithm is based on the tabu search. The tabu search is a global optimization metaheuristic that can be used to solve combinatorial optimization problems. Basically it consists of several elements called the initial solution, neighborhood, searching strategy, move, tabu list, aspiration function, and stopping criteria. The neighborhood of a solution is the set of all possible solutions which can be accessed from the current solution by a move. The move is a function which transforms a solution into another solution. At each step the neighborhood of the current solution is searched in order to find an appropriate neighbor which can be found inexpensively. To avoid cycling (trapped in a local optimum) and to add robustness of the search, forbidden moves are stored in a short term memory called a tabu list. A move is identified to store in the tabu list as a way to prevent future moves that would undo the effects of previous moves. Nevertheless, a forbidden move can be performed while it suffices requirements of aspiration functions. The stopping criteria may be a limit on the execution time, number of iterations, number of iterations without improvements, or criterion performance.

Tabu search is a general approach for finding a near optimal solution of combinatorial optimization problems. Its every application needs detailed definitions of basic elements (e.g., initial solution, neighborhood, searching strategy, move, etc.) and values of several tuning parameters such as the size of tabu list, level of aspiration, stopping criteria, etc. These elements and parameters interact to determine the speed of convergence, performance, and running time.

The tabu search scheme for clustering[2] can be stated as follows.

1. start with an initial solution and evaluate its objective function value.
2. Generate randomly trial solutions from the current solution and evaluate their objective function values.
3. If the best of these trial solutions is out the tabu list, then it is considered to be the new solution.
4. If the best of these trial solutions is in the tabu list but its value of objective function is smaller than one of the current solution, then it is also considered to be the new solution.
5. If both conditions stated in steps 3 and 4 are not satisfied, then remove the best trial solution and go to step 3 unless the set of trial solutions is empty.
6. If the set of trial solution is empty, then go to step 2.
7. Update the tabu list if necessary.
8. If the new solution is closer to the optimal solution than the best solution so far, then the new solution becomes the best solution.
9. The new solution becomes the current solution. Go to step 2 until the specified number of iterations is reached.

The tabu search is executed for a certain number of iterations. A feasible solution is obtained after one iteration. On termination, the best solution obtained so far is the solution obtained by the algorithm. The obtained solution is not the optimal solution. The purpose of the tabu search is to shorten the distance between the obtained solution and the optimal solution as small as possible within these iterations.

It is clear that trial solution generation is the key component of the tabu search for clustering. Our improved algorithms focus on the trial solution generation. The proposed method(2) is a trivial method. A trial solution is generating from the current solution. For each point, its group allocation is examined. The determination of the examination is according to the value of a random number generating from a uniformly random number generator $u(0, 1)$. If the value of the generating random number is less than the probability threshold P, then the group allocation of the point is kept. Otherwise, the group allocation of the point is changed. Its new group is assigned also by a random number generator. The method is called probability threshold.

The group allocation is the main drawback of the probability threshold method. The historical information about group allocation of a point is not used to determine the new group for the point. In this manner, the group allocation of a point is changed even if its group allocation is correct. Such a wrong change degrades the effectiveness of the generating trial solution.

To avoid the degradation, a P-decay method is proposed to determine whether the group allocation is change or not. In this manner, the probability for the group allocation of a point to be changed decreases gradually P ratio. In the last generating trial solutions, the probability for the group allocation of a point to be changed is q. If the group allocation of the point is not changed, then the probability will become Pq. If the group allocation of the point is changed, then the probability is reset to 100%. Using the method, a correct group allocation can be kept and a wrong group allocation can be changed eventually.

In the P-decay method, the probability for the group allocation of a point to be changed is computed when the new solution is obtained. The storage space of each point has a field to record the probability. While the current solution is replaced with the new solution, the group allocation of each point is compared and the probability of each point is computed. In this manner, only little memory space is needed to store the information about probability of each point. The computation of the probability is also little. Therefore the extra cost of the probability maintenance is little in the P-decay method. The effectiveness of generating trial solutions and the quality of the solution both are increased by the P-decay method.

The correct group allocation of a point may be changed in the P-decay method. This phenomenon should be eliminated to increase the effectiveness of generating trial solutions. An improved method is proposed to cancel the spot. The probability for the group allocation of a point to be changed is proportional to the distance between the point and its center. Therefore the group allocations of points that are far away their centers are necessary to be changed to improve the efficiency of the tabu search algorithm.

The value of the distance is determined with the following fact. If points are uniformly distributed within a circle with radius R, then the average distance between points and the center of the circle is $\frac{R}{\sqrt{2}}$. Distances between points, that are within a group, and the group's center are computed. The average distance, d, of points with the group can be obtained. For each point within the group, its distance is compared with $\sqrt{2}d$. If its distance is greater than $\sqrt{2}d$, then the group allocation of the point is reassigned. Otherwise, there are no modification for the point. This strategy is called the distance proportional method.

In the distance proportional method, the group allocation of a point is randomly assigned while it must be changed. Such a randomized modification moves out a local minima. The correct group allocations can be kept. The effectiveness of generating trial solutions is improved by this issue. The wrong group allocations can be changed eventually. Therefore the optimal solution should be found by the method. The quality of the solution is also improved in a certain number of iterations.

Although the effectiveness of trial solution generation is improved dramatically by distance proportional method, the speed of convergence seems to be further enhanced. It is a drawback in the distance proportional method to reassign randomly farther points from its group center. This drawback can be eliminated by that each point of these points is assigned to a group which is nearest to it. Therefore for a group there are three areas surrounding its center. Points in the internal area are fixed in the group to keep the effects of previous works. Points in the middle area are reassigned randomly to jump local minimum solution. Points in the external area are reassigned to the nearest group to improve the speed of convergence.

4. Experimental studies

In this paper, an algorithm for the clustering problem is presented. The algorithm is based on the tabu search technique. There are some strategies are proposed for generating trial solutions. In our experiments, these strategies will be compared with the probability threshold method. These methods have been coded and executed on an IBM compatible machine on several test problems.

	probability threshold number of trial solutions			*P*-decay method number of trial solutions		
	20	25	30	20	25	30
A_1	43602.64	36483.32	33273.39	25798.65	24607.50	24181.57
B_1	48503.79	31671.93	32743.95	27949.84	26055.1	25859.68
C_1	44349.33	38133.13	34492.19	27907.87	31104.01	27266.27
D_1	40270.77	37614.86	31978.78	27233.00	24044.15	23958.51
E_1	29661.16	28005.64	24785.51	20634.42	19698.62	19380.42
F_1	29217.19	27262.22	24531.52	20123.56	20117.89	20058.77
G_1	28451.19	26554.23	26006.58	23065.19	19681.35	19677.57
H_1	30293.52	26671.51	26202.72	20364.77	19880.08	19549.44

Table 1: The objective function values

The testing data is randomly generated in R^2. There are three classes of testing data. They are overlapped clusters, touching clusters, and separated clusters. For each class, there are at least three testing cases whose numbers of points are different. The number of points in a case may be 16 , 64, or 256 and the number of groups is 5. For each case, the experiment is repeated at least ten times. Only the average result for a case is shown in our experimental result tables.

	probability threshold number of trial solutions			*P*-decay method number of trial solutions		
	20	25	30	20	25	30
A_1	1691	2362	3196	1839	2664	3462
B_1	1706	2389	3222	1867	2694	2220
C_1	1727	2401	3236	1895	1059	1451
D_1	1747	2420	3253	1920	2767	3572
E_1	1795	2616	3394	1842	2653	3468
F_1	1817	2639	3418	1879	2704	3484
G_1	1836	2663	3445	1903	2740	3549
H_1	1806	2680	3466	1936	2774	3588

Table 2: Computation time (unit: seconds)

The number of trial solutions that is generated in an iteration is related to the efficiency of the algorithm. If the number of trial solution is large, then the computational time of the iteration in the algorithm is huge. On the other hand, if the number of trial solution is large then the probability for finding a solution that is better than the best solution so far is also increased. Therefore the relations between the objective function value, the number of trial solution, and computation time are studied. The experiment is performed by the probability threshold method and the *P*-decay method.

The relations between the objective function value and the number of trial solutions for these methods are listed in Table 1. For each case, the objective function value is smaller when its number of trial solution is larger. In each case, the objective function value of the *P*-decay method with 20 trial solution is better than the one of the probability threshold method with 30 trial solutions. The relations between the number of trial solutions and the computation time are listed in Table 2. The computation time is longer when its number of trial solutions is larger. In each case, the computation time of the *P*-decay method with 20 trial solution is less than the one of the probability threshold method with 30 trial solutions. The *P*-decay method is better than the probability threshold method in the sense that the better objective function value is obtained in less computation time.

	P-decay method		distance proportional	
	values	time (sec)	values	time (sec)
A_2	356574.9	13090	143970.8	13982
B_2	357636.1	13171	147960.3	14080
C_2	307897.5	13215	155197.5	16507
D_2	301357.7	13093	138765.6	16587

Table 3: Experimental results of the *P*-decay method and the distance proportional method

The performances of the *P*-decay method and the distance proportional method are tested. The result is listed in Table 3. The objective function values of the distance proportional method are better than the ones of the *P*-decay method. The computation time of the distance proportional method is little larger than the one of the *P*-decay method. In the distance proportional method, the objective function value converges to its optimal value very quickly. For these cases in Table 3, the number of iterations of the distance proportional method whose objective function values are better than ones of the *P*-decay method are 12, 13, 15, and 16, respectively.

	distance proportional method				enhanced distance proportional method		
	50 iterations		300 iterations				
	values	sec	values	sec	values	sec	iterations
A_3	67422.53	158	57119.51	1049	36470.23	168	10
B_3	68703.25	158	65109.66	1050	51427.19	116	9
C_3	102346.4	158	72517.33	1050	62196.26	54	6
D_3	65899.35	258	57242.18	1686	33113.04	366	30
E_3	68859.73	258	60565.78	1686	51180.61	190	23
F_3	99435.14	258	76145.57	1686	62196.25	110	6
G_3	69672.62	373	57210.73	2423	33011.48	95	12
H_3	67085.64	374	58455.67	2421	51180.63	376	22
I_3	96385.59	375	67654.02	2431	62196.25	68	5

Table 4: The experimental results of the distance proportional method and its enhanced method

The comparison between distance proportional method and its enhanced method is listed in Table 4. The stopping criterion is that the total number of iterations without improvements reaches 2. The number of iterations for the enhanced method is so small that the speed of convergence is quick. From the values of the objective function, solutions obtained by the enhanced method is better than one by the distance proportional method. The computational time of the enhanced method is about from 1 to 7 minutes. That is, the enhanced method can be used in practical systems.

5. Conclusions

In this paper, we have developed an improved algorithm for solving the clustering problem that is based on the tabu search technique. The algorithm has been implemented and tested on various problems. Some smart strategies for generating trial solutions are proposed to improve its effectiveness. From the experiments, these strategies are shown to improve the efficiency of the algorithm and the quality of the solution. The preliminary computational experience is very encouraging. The initialization of clusters[1] is an interesting problem to work in the future. The intensification and diversification actions in our algorithm are necessary to further study.

References

1. M. B. Al-Daoud and S. A. Roberts, New methods for the initialisation of clusters, Pattern Recognition letter, 17 (1996), pp 451-455.
2. K. S. Al-Sultan, A tabu search approach to the clustering problem, Pattern Recognition, 28(9), 1995, pp 1443-1451.
3. M. R. Anderberg, Cluster analysis for application, Academic Press, New York.
4. R. Dubes and A. K. Jain, Clustering techniques: the user's dilemma, Pattern Recognition, 8, 1976, pp 247-260.
5. R. O. Duda and P. E. Hart, Pattern classification and scene analysis, John Wiley & Sons, New York, 1973.
6. E. Forgy, Cluster analysis of multivariate data: efficiency versus interpretability of classifications, Biometrics, 1965, pp 768.
7. D. E. Goldberg, Genetic algorithms: search, optimization and machine learning, Addison-Wesley, Reading, MA.
8. M. Ismail and M. Kamel, Multidimensional data clustering utilization hybrid search strategies, Pattern Recognition 22(1), 1989, pp 75-89.
9. S. Z. Selim and M. A. Ismail, K-means-type algorithm: generalized convergence theorem and characterization of local optimality, IEEE Trans. Pattern Anal. Mach. Intell., Vol. 6, 1984, pp 81-87.
10. R. A. Johnson and D. W. Wichern, Applied multivariate Statistical analysis, Prentice-Hall, Englewood Cliffs, NJ.
11. L. Kaufman and P. Rousseeuw, Finding groups in data: an introduction to cluster analysis, Wiley, New York, 1990.
12. H. Spath, Cluster analysis algorithms, Ellis Horwood, Chichester, UK.