Journal of Animal and Veterinary Advances 4 (5): 535-539, 2005

Auto-K Dynamic Clustering Algorithm

Xiwu Han and Tiejun Zhao
School of the Computer Science and Technology, Harbin Institute of Technology, Harbin City 150001, China

Abstract: Most clustering methods need a pre-determined clustering number or a certain similarity threshold, which makes them dependent on heuristic knowledge. The X-means method tries to estimate the number of clusters but only converges locally. This paper presents a novel and simple clustering algorithm named as Auto-K after its descriptive parent-algorithm-K-means, though Auto-K theory can be generalized beyond certain given deriving algorithms. In Auto-K, the algorithm itself automatically selects a globally optimal clustering number for the involved population, by maximizing the clustering fitness and thus the clustering process can be said to be really dynamic and most accordant with human's common sense in clustering.

Key words: Clustering, Intracluster similarity, Intercluster similarity, Clustering fitness

Introduction

Clustering methods are widely involved almost in all the fields of AI, especially in data mining, results of which often serve as the necessary foundational knowledge for natural language processing, machine translation and other more practical applications, such as store-customer grouping, web-document categorizing, genes and protein classifying, so on and so forth. According to [JD1988, KR1990 and PAS1996], clustering in data mining is a discovery process that groups a set of data such that the intracluster similarity and intercluster dissimilarity are maximized and the intercluster similarity and intracluster dissimilarity are minimized. Since the resulting clusters will be used to deal with the real changeable world, people concerned tend to hold a dynamic viewpoint while developing an algorithm for this problem. Almost all the so-called dynamic clustering algorithms existing nowadays, such as K-means [JD1988], PAM [KR1990], CLARANS [NH1994], DBSCAN [EKSX1996], CURE [GRS1998], ROCK [GRS1999] and CHAMELEON [KHK1999], are designed to handle incremental samples in a certain database. However, none of them are really dynamic in that a pre-determined clustering number or a certain similarity threshold is often indispensable in the conditional settings of all the algorithms. And because a reasonable clustering number or similarity threshold can never be easily determined without a thorough investigation into the involved population, the above algorithms are, respectively in a certain degree, still confined to a static modeling disadvantage. There have been some efforts made to overcome this disadvantage, e.g. the X-means method [PK2000], which tries to estimate the number of clusters but only converges locally.

In this paper, we will present a novel and much simpler clustering algorithm named as Auto-K after its descriptive parent-algorithm—K-means, though we will state that Auto-K theory can be generalized or be independent from certain given deriving algorithms. In Auto-K, the algorithm itself automatically and globally selects an optimal clustering number for the involved population, by maximizing the clustering fitness and thus the clustering process can be said to be really dynamic and most accordant with human's common sense in clustering.

Materials and Methods

Generally, techniques used widely in clustering can be grouped into partitional or hierarchical classes, with K-means (JD1988), PAM (KR1990), CLARANS (NH1994), etc. as examples for the former and CURE (GRS1998), ROCK (GRS1999), CHAMELEON (KHK1999), etc. on the behalf of the latter. Partitional clustering methods attempt to break a population of data into k clusters such that the partition optimizes a given criterion (JD1988, KR1990, NH1994, CS1996 and KHK1999). One typical example is the K-means algorithm, in which the number of clusters k for a training data set is pre-determined by the user and then the resulting model is used to group incremental data in the same field (JH1997). Hierarchical clustering algorithms produce a nested sequence of clusters, with a single all-inclusive cluster at the top and singleton clusters at the bottom (KHK1999). Ostensibly, detailed tricks applied in hierarchical clustering vary widely, yet they all involve the bottom-upward merging process, which stops when the desired number of clusters is obtained or the distance between two closest clusters is above a certain threshold.

Both kinds of clustering algorithms are intended to deal with incremental data dynamically, but they may have ignored the fact that newly input items may well locate beyond the domain of the training samples. And the new data may well in turn have a vital influence on the clustering number or the stopping condition. Let's have a look into human's

common sense in clustering. For example, with k = 2, the resulting clusters for a population of {carp, frog, crucian, crocodile, chub, sardine} may be {(frog, crocodile), (crucian, carp, chub, sardine)} as amphibian and fish. If a new item is "whale", it may fall into the second cluster as a kind of fish. However, if another new one "seal" is added, there occurs the problem, for "whale" and "seal" tend to form a new cluster as sea mammal. Therefore, items that can be correctly classified will never go beyond the domain of the training samples until the clustering number or threshold can be automatically changed or determined.

A recent clustering method, the X-means algorithm (PK2000), tries to overcome the limitations of K-means by estimating the number of clusters. This method goes into action after each run of K-means, making local decisions about which subset of the current centroids should split themselves in order to better fit the population. Although the splitting decision is done by computing the Bayesian Information Criterion with very complicated formulas, it's made only according to the local situation within the cluster intended to be split. Therefore, X-means can only converge at a locally optimal value of K and partially solve the problem.

Auto-K Algorithm: Auto-K algorithm can automatically select a globally optimal clustering number for the involved population while clustering it. In this paper we will describe the novel algorithm via the most common clustering method-K-means, though it should be noticed that Auto-K theory can be generalized or be independent from certain given deriving algorithms.

The K-means algorithm is non-hierarchical. Usually, it at first takes k items from the population as centroids for the k final required clusters. In this very step the initial centroids are often chosen such that they are mutually farthest dispersed. Next, the algorithm examines each item in the population and assigns it to one of the k clusters depending on the minimum distance between it and the respective centroid. And the centroid's position is recalculated every time an item is added to the cluster and this continues until all the components are grouped into the final required number of clusters. The number k is often determined according to some heuristic information previously learned about the domain. However, this information may serve as a double-edged sward, which means it may become quite harmful if given too much weight. And consequently the clustering result may not be the best one.

Therefore, to determine the best k and the optimal clustering result, we need to integrate into the K-means algorithm an evaluating technique that finds out how appropriate the clustering result is. According to [JD1988, KR1990 and PAS1996], the best clustering result should maximize the intracluster similarity and intercluster dissimilarity and minimize the intercluster similarity and intracluster dissimilarity. The intracluster similarity can be quantified via some function of the reciprocals of intracluster radiuses within each of the resulting clusters and the intercluster similarity can be quantified via some function of the reciprocals of intercluster radiuses of the clustering centroids. These kinds of radiuses are illustrated in Fig. 1.

Definition 1: The intracluster similarity of a cluster C_i (1 = i = k), denoted as $S_{tra}(C_i)$, is defined by

$$S_{tra}(C_i) = \frac{1+n}{1+\sum_1^n dist(I_j, Centroid)} \tag{1}$$

In here n is the number of items in cluster C_i, 1 = j = n, I_j is the jth item in cluster C_i and *dist* (I_j, *Centroid*) calculates the distance between I_j and the centroid of C_i, which is the intracluster radius of C_i, either in terms of Euclidean or Mahalanobis or others. 1 is added to the denominator and the numerator to smooth the value of $S_{tra}(C_i)$ and allow for possible singleton clusters.

Definition 2: The intracluster similarity for one of the possible clustering results C, denoted as $S_{tra}(C)$, is defined by

$$S_{tra}(C) = \frac{\sum_1^k S_{tra}(C_i)}{k} \tag{2}$$

In here k is the number of resulting clusters in C.

Definition 3: The intercluster similarity for one of the possible clustering results C, denoted as $S_{ter}(C)$, is defined by

$$S_{ter}(C) = \frac{1+k}{1+\sum_1^k dist(Centroid_j, Centroid^2)} \tag{3}$$

In here k is the number of resulting clusters in C, $1 = j = k$, $Centroid_j$ is the centroid of the j th cluster in C, $Centroid^2$ is the centroid of all centroids of clusters in C and $dist$ ($Centroid_j$, $Centroid^2$) calculates the distance between $Centroid_j$ and $Centroid^2$, which is the intercluster radius of $Centroid_j$, either in terms of Euclidean or Mahalanobis or others. 1 is added to the denominator and the numerator to smooth the value of $S_{ter}(C)$ and allow for possible all-inclusive clustering result.

Now the best k and the optimal clustering result C^* could be availed by maximizing $S_{tra}(C)$ and minimizing $S_{ter}(C)$, but the fact is that more often than not the maximum of $S_{tra}(C)$ and the minimum of $S_{ter}(C)$ can not be achieved with the same clustering result and on the contrary there is usually a tradeoff needed between the two. Therefore, we combined $S_{tra}(C)$ and $S_{ter}(C)$ into one single criterion — clustering fitness to make the algorithm more practical and convenient.

Definition 4: The clustering fitness for one of the possible clustering results C, denoted as CF, is defined by

$$CF = \lambda \times S_{tra} + \frac{1-\lambda}{S_{ter}} \tag{4}$$

In here $0 < \lambda < 1$ is an experiential weight. For example, with $\lambda = 0.5$, $S_{tra}(C)$ and $S_{ter}(C)$ will impose the same influence on the value of CF.

Thus the problem can be solved by simply maximizing the value of CF. And for a new population to be clustered, which includes N items, the Auto-K algorithm can be briefly described as follows.

For $1 = k = N$, do

a. Select k initial cluster centroids: $Centroid_1$, $Centroid_j$, …, $Centroid_k$ ($1 = j = k$).

b. Cluster the population via K-means method with k and output the clustering result C.

c. Calculate the CF for C via formulas (1), (2), (3) and (4).

End do

Take C^* = ArgMax(CF) as the optimal clustering result.

Experimental Analysis: For the above-mentioned algorithm we performed a simple experiment on the points as shown in Fig. 1 and the relative coordinate locations are listed in Table 1. In the experiment, distances were calculated in term of Euclidean measure, initial centroids were selected randomly yet as dispersed as possible and ***ë*** was set to be 0.5. Table 2 gives the clustering results.

In the fifth column of Table 2, we list the values of $1/ S_{ter}(C)$, which in fact is the intercluster dissimilarity, for the convenience of readers' comparison. As we can see, there is no linear relationship either between k and $S_{tra}(C)$, or between k and $1/S_{ter}(C)$. Among the clustering results, C_7 achieves the maximum $S_{tra}(C)$, C_4 gives the minimum $S_{ter}(C)$ and the clustering fitness of C_4 is also the maximum. Hence, 4 is the best k and C_4 is the optimal clustering result for this population, as marked with asterisks in Table 2.

The computational complexity of the Auto-K algorithm, however, is much higher than the K-means method. The complexity of K-means is estimated as O(N) and that of Auto-K as O(N^2). Yet the high complexity can be reduced by selecting only a few most possible candidates for k. For example, in this experiment some good candidates may be k = {2, 3, 4}.

Table 1: Population for the Experiment

a	b	c	d	e	f	g
(1, 1)	(5, 2)	(6, 3)	(2, 4)	(3, 5)	(2, 6)	(1.5, 5)

Table 2: Auto-K Clustering Results

k	Initial Centroids	Clustering Results	$S_{tra}(C)$	$1/S_{ter}(C)$	CF
1	f	{(a, b, c, d, e, f, g)}	0.48	0.50	0.98
2	c, f	{(a, b, c) (d, e, f, g)}	0.83	1.50	2.33
3	a, c, f	{(a) (b, c) (d, e, f, g)}	1.45	2.21	3.66
4*	a, b, c, f	{(a) (b) (c) (d, e, f, g)}*	1.80	2.34	4.14*
5	a, b, c, d, f	{(a) (b) (c) (d) (e, f, g)}	1.85	2.15	4.00
6	a, b, c, d, e, f	{(a) (b) (c) (d) (e) (f, g)}	1.79	2.08	3.87
7	a, b, c, d, e, f, g	{(a) (b) (c) (d) (e) (f) (g)}	2.00	2.08	4.08

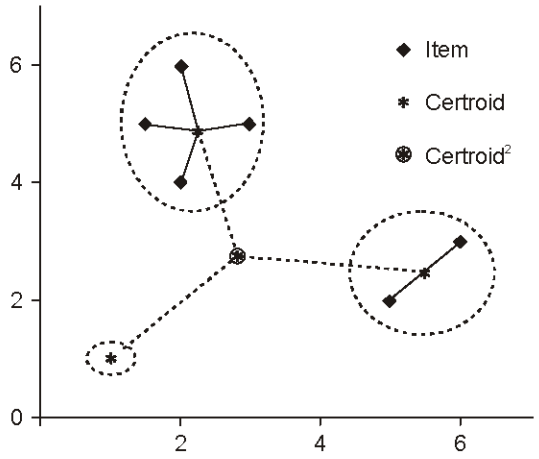

Fig. 1: Intracluster and intercluster radiuses

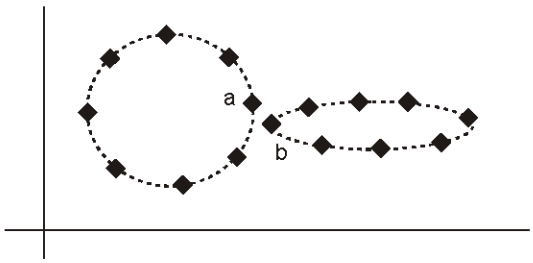

Fig. 2: Shapes of clusters are important

Generalized Auto-K Theory: The foundational theory of Auto-K is independent from the K-means algorithm or any other clustering methods. It offers an objective and quantifiable evaluating measure for any clustering results. From such a point of view, clustering can be regarded as a searching process for the optimal result in the whole space of clustering hypotheses and the Auto-K theory itself can fulfill such a task by exhaustively searching the hypothesis space. But a fatal disadvantage is the large space, the total number of hypotheses of which is the sum of Stirling numbers of the second kind for $1 = m = N$. For instance, if $N = 10$, there will be 25,104 kinds of possible clustering results to be searched. Therefore, the Auto-K theory is better to be integrated into other clustering methods for a practical reason.

To achieve desirable clustering results with Auto-K theory, it is very important to describe appropriately the items in the population. Generally, for $\boldsymbol{x}$-dimensional data that cannot be clustered properly via Auto-K, $(\boldsymbol{x} + 1)$-dimensional description may well improve the situations. For example, the shapes of clusters tend to be worth consideration in some clustering problems, such as the one illustrated in Fig. 2.

With 2-dimensional description, item a and item b will surely be clustered into the same class. If the angular coordinate positions are also took into consideration, making the description 3-dimensional, item a and item b will be correctly clustered into different classes.

Conclusion

We have described a kind of Auto-K algorithm via K-means and a simple experiment shows that this method is able to find out the best k and the optimal clustering result, thus saving the necessary investigation into an unfamiliar population before the number of anticipated clusters or a clustering similarity threshold is determined. Although K-means tends to be greedy in its searching manner when enlarging a possible cluster, the globally optimal result can still be achieved by simply performing more times K-means with different initial centroids. Moreover, Auto-K theory can also benefit hierarchical clustering methods by evaluating clustering results at different levels.

Most clustering methods depend too much on previously learned knowledge about the population, which is needed to pre-determine a number of possible clusters or a threshold for stopping the clustering procedure. Sometimes this may give too much weight to the heuristic information and make the algorithm suffer from overfitness. On the contrary, there will never occur such a problem with Auto-K theory because it thoroughly depends on the observation and description of the concerned population and there is very little if any heuristic information needed.

The problem with Auto-K theory is the large clustering cost. It makes no use of experiences on similar populations, while such knowledge may curtail the computing time and searching space. Some of our future work will focus on finding a suitable tradeoff between the effectiveness and efficiency for Auto-K clustering methods.

References

Jain, A. K. and R. C. Dubes, 1988. *Algorithms for Clustering Data*. Prentice Hall, 1988.

Kaufman L. and P. J. Rousseeuw, 1990. *Finding Groups in Data: an Introduction to Cluster Analysis*. John Wiley & Sons,

Hubert, L. J., P. Arabie and G. De Soete, 1996. *Clustering and Classification*. World Scientific,

Ester, M., H. P. Kriegel, J. Sander and X. Xu, 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. of the Second Int'l Conference on Knowledge Discovery and Data Mining, Portland, OR.

Pelleg, D., A. Moore, 2000. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In: Proceedings of the Seventeenth International Conference on Machine Learning. San Francisco.

Sudipto Guha, Rajeev Rastogi and Kyuseok Skim, 1998. CURE: An efficient clustering algorithm for large datadases. In: Proc. of 1998 ACM-SIGMOD Int. Conf. On Management of Data.

Sudipto Guha, Rajeev Rastogi and Kyuseok Skim, 1999. ROCK: arobust clustering algorithm for categorical attributes. In: Proc. of the 15th Intl. Conf. on Data Eng.

Jiarong Hong, 1997. Inductive Learning. 81-82. Science Press House, China.

Karypis, G., E. H. Han, V. Kumar, 1999. CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling. In the Proc. of IEEE Computer.

Ng, R. and J. Han, 1994. Efficient and effective clustering method for spatial data mining. In: Proc. Of the 20th VLDB Conference, Santiago, Chile, pp: 144-155.