Sample Index Based Encoding for Clustering Using Evolutionary Computation

Xiang Yang[1,2] and Ying Tan[2]

[1] AVIC Leihua Electronic Technology Research Institute, Wuxi 214063, China
[2] Key Laboratory of Machine Perception (Ministry of Education), Peking University,
Department of Machine Intelligence, School of EECS, Peking University,
Beijing 100871, China
{yangxiang,ytan}@pku.edu.cn

Abstract. Clustering is a commonly used unsupervised machine learning method, which automatically organized data into different clusters according to their similarities. In this paper, we carried out a throughout research on evolutionary computation based clustering. This paper proposed a sample index based encoding method, which significantly reduces the search space of evolutionary computation so the clustering algorithm converged quickly. Evolutionary computation has a good global search ability while the traditional clustering method k-means has a better capability at local search. In order to combine the strengths of both, this paper researched on the effect of initializing k-means by evolutionary computation algorithms. Experiments were conducted on five commonly used evolutionary computation algorithms. Experimental results show that the sample index based encoding method and evolutionary computation initialized k-means both perform well and demonstrate great potential.

Keywords: Clustering, evolutionary computation, sample index based encoding, initializing k-means.

1 Introduction

Clustering refers to partitioning data into different clusters according to their similarities. It is an unsupervised machine learning method that does not require the pre-specified categories information. It directly mining unlabeled data according to their inherent structure, and automatically partitions them into several clusters based on the intrinsic properties of the data. The goal of clustering is to get the greatest similarity between samples of the same cluster and the smallest similarity between samples of the different clusters. Clustering has been widely used in many fields. For example, in the field of text processing, document clustering can effectively organize different documents together based on subject, which can help people to filter the documents and find what they needed quickly.

There are many traditional clustering algorithms, such as k-means [1], hierarchical clustering [2] and expectation maximization (EM) [3]. The target of k-means algorithm is to get the smallest sum of distances within the same classes.

Y. Tan et al. (Eds.): ICSI 2014, Part I, LNCS 8794, pp. 489–498, 2014.

It will get a satisfying result through iterative updating the center of each cluster and updating the cluster each sample belongs to. However, using such a clustering approach always has the risk of falling into local optimum.

Evolutionary computation is a kind of optimization algorithm by simulating the natural biological process of evolution. Evolutionary computation algorithms generally maintain a population of solutions, and gradually improve the quality of the solution through the evolution of the population. There are some common evolutionary computation algorithms, like genetic algorithms [4], evolutionary strategies [5], differential evolution [6], particle swarm optimization algorithm [7] and firework algorithms [8]. Evolutionary computation has incomparable superiority compared to traditional optimization algorithms. It does not need to calculate the gradient of the objective function for achieving the optimization, and it has strong robustness, not easy to fall into local optimum.

In recent years, evolutionary computation is used by many researchers to improve the quality of clustering. Ujjwal Maulik et al. used the genetic algorithm to optimize the inter-clustering distance [9,10], Das, S. et al. and Paterlini, S. et al. applied differential evolution to the clustering problem [11,12], while Van der Merwe, DW et al. and Chen, ChingCYi et al. studied the particle swarm optimization based clustering [13,14].

In this paper, we first reviewed the basic principle and process of using evolutionary computation to cluster, and then we proposed a sample index based encoding method, this method can effectively reduce the search space of evolutionary computation algorithms, which will make the clustering algorithm converge quickly. Finally we focused on the study of initializing k-means by evolutionary computation algorithms.

This paper is organized as follows: Chapter 1 is introduction. Chapter 2 gives a brief introduction to k-means and the clustering based on evolutionary computation. Chapter 3 presents a sample index based encoding method. Chapter 4 demonstrates the using of evolutionary computation to initialize k-means. The experimental results are given in Chapter 5. Chapter 6 makes the summaries.

2 Formulated Description of Clustering

Given a data set D, whose amount of the samples is N and the dimension of each sample is d. $D = x_1, x_2, ..., x_N$ where x_i represents a d-dimensional vector, $i = 1, 2, ...N$. Clustering algorithms require these N samples be partitioned into K clusters. Many clustering algorithm use the centroids of the clusters to determine the cluster attribution. Assuming the centroid of the i-th cluster is c_i, $i = 1, 2, ..., K$. For the sample x_j, clustering algorithm will calculate the distance between x_j and all the centers of K clusters, and partition x_j into the k-th cluster if the distance between x_j and the centroid of the k-th category is the smallest. The process can be represented mathematically by the following formula:

$$k = \arg\min_{i} \|x_j - c_i\|^2 \tag{1}$$

where $\|x_j - c_i\|^2$ represents the Euclidean distance between the two vectors.

The goal of clustering is to make the samples inside the same cluster have the greatest similarity. This goal can be achieved by minimizing the within-cluster sum of squares (WCSS). The definition of WCSS is as follows [9]:

$$\mathrm{J} = \sum_{i=1}^{K} \sum_{x_j \in C_i} \|x_j - c_i\|^2 \tag{2}$$

where C_i represents a collection of samples in class i, the WCSS represents the sum of all samples distances to their corresponding clusters centroid.

K-means applies an iterative way to update the centroid of the cluster to obtain a promising clustering result. It first randomly initialize a centroid for each cluster, then each samples cluster label is determined according to Formula 1, then the average of all samples within the cluster i is set as the new centroid of the i-th cluster, as shown in the following equation [1]:

$$c_i = \frac{\sum_{x_j \in C_i} x_j}{|C_i|} \tag{3}$$

After obtaining the new centroid for each cluster, k-means process the clustering according to Formula 1 again, and then get new cluster centroids. This process is iterated until the termination condition is satisfied.

The clustering result of k-means is susceptible to the initial cluster centroid. If the selection of initial cluster centroid is not good, k-means is easy to fall into local optimum. While the evolutionary computation has excellent ability of global optimization. Therefore using evolutionary computation to cluster data can get a better quality of clustering.

When using evolutionary computation to clustering, we have to encode the way of clustering into individuals. The individual is represented by a multi-dimensional vector and the way of encoding has significant impact on the clustering results. There two common ways used for encoding: cluster centroid based encoding [9] and sample category based encoding [15].

When using cluster centroid based encoding, all clusters centroids will be joined into a single vector, as an individual in evolutionary computation, which is represented as $< c_1, c_2, , c_K >$.

Assuming a data set has 1000 samples in total, and they can be partitioned into 20 clusters, each dimension of the data is 10, the clustering problem will be encoded as a 200-dimensional vector. The vector is represented by the conjunction of 20 10-dimensional vectors, the 20 vectors stand for the 20 centroids for all the 20 clusters.

Given the centroid of each category, each sample is partitioned to the corresponding cluster according to Formula 1, and then use the formula Formula 2 to get the WCSS, which will then be used as an individuals fitness function.

Sample category based encoding method encodes each sample's cluster directly. Each individuals dimension equals to the number of samples, the individual is expressed as $< L_1, L_2, , L_N >$, where L_j stands for sample x_js cluster label, $j = 1, 2, ..., N$, L_j ranges between 1 to K as an integer. If the j-th sample belonging to the category i, $i = 1, 2, ..., K$, then the individual value of the j-th dimension L_j is i.

After given each samples category, the center of each class is calculated by using the Formula 3, then the WCSS will be used as an individuals fitness function.

3 Sample Index Based Encoding

In order to further improve the quality of clustering, we propose to use the sample index based encoding method. Cluster centroid based encoding using a point in d-dimensional space to represent a cluster centroid, so each individuals dimension after encoding is $K * d$. Sample index based encoding using the sample from the sample set as a cluster centroid; we just have to record the samples index in the sample set. Therefore the individuals dimension after encoding is K. The individual is represented as $< I_1, I_2, , I_K >$.

Under such encoding method, the i-th clusters centroid is the I_i-th sample in the sample set x_{I_i}. After each clusters centroid is determined, each samples cluster is calculated by using Formula 1, then the WCSS will be used as an individuals fitness function.

Assuming a data set has 1000 samples in total, and they can be partitioned into 20 clusters, each dimension of the data is 10, the clustering problem will be encoded as a 20-dimensional vector. If the 10-th dimension of the vector is 426, the centroid of the 10-th cluster is the 426-th sample in the sample set.

The individuals dimension of cluster centroid based encoding is $K * d$, the individuals dimension of sample category based coding is N, while the individuals dimension of sample index based encoding is K. Generally K is much less than $K * d$ and N. Therefore the dimension of the proposed encoding method is much lower than the other two ones. The relationship between the search space of evolutionary computation and the individuals dimension is exponential, so the low-dimensional encoding means you can significantly reduce the search space. After the reduction, evolutionary computation algorithms can easily find the optimal solution and get a better result.

General speaking, before using evolutionary computation to search for the optimal solution, we need to specify the upper and lower bounds for each dimension in the search space. The upper and lower bounds of the cluster centroid based encoding is determined by the range of training data. For example the range of c_is x-th dimension should be equal to the range of the x-th dimension of all the data. So actually the process of evolutionary computation is to search a solution within a hypercube. But in general the data are not evenly distributed in the hypercube, the data may be concentrated in certain areas, and most areas in the search space don't have any data. The evolutionary computation algorithm will inevitably enter areas without data to search. This will waste a lot of time.

When using sample index based encoding, since each centroid comes from the sample set, the selection of centroids is more close to the real distribution of the data. So evolutionary computation algorithm will not enter the areas without data, but it will only search within areas of data. This mechanism ensures that the evolutionary algorithms search time is spent where it makes sense, thus evolutionary algorithm can quickly converge to the optimal solution, resulting in better clustering results.

4 K-means Initialized by Evolutionary Computation

Evolutionary computation is well known for its excellent global search capability. Evolutionary computation algorithms adopt the stochastic strategy to avoid trapping into the local optimum. Individuals have certain probability to jump out of current searching areas, which enables the evolutionary computation algorithms to explore the global optimum in the whole search space. But such algorithms may not perform well when searching the local area to further improve the current best solution. The stochastic way of local search cannot finely guide the current best solution to the actual best solution near it.

K-means minimizes the WCSS by iterating the following procedures: updating the cluster label of each sample according to the centroids of clusters, and then updating the centroids of clusters according the cluster label of each sample. The updating of centroids of clusters at successive iterations takes place in the local area; the centroids at the next iteration are not far from the centroids at the previous iteration. Therefore k-means has strong local search capability. But such searching strategy cannot explore the whole search space sufficiently, leading to a poor global search capability. If the initial centroids are not well chosen, k-means will trapping into the local optimum.

In order to exploit the synergy of global search ability and local search capability, the combination of evolutionary computation with k-means have been studied. For example Ahmadyfard, A. et al. combined particle swarm optimization algorithm and k-means algorithm to to get a better clustering algorithm [16].

In this section we combine the proposed sample index based encoding method with k-means to study their synergy. First we use an evolutionary computation algorithm to get K centroids. Evolutionary computation is able to obtain quite good centroids over the whole search space due to its excellent global search capability. But these centroids need to be further tuned in the local area to improve the clustering performance. We use k-means to tune these centroids by taking these centroids as the initial centroids of k-means. k-means will exploit the local area to search for better centroids in an iterative way. k-means initialized by evolutionary computation combines the strength of evolutionary computation and k-means, leading to both excellent global search capability and excellent local search capability.

The procedure of k-means initialized by evolutionary computation are shown in Algorithm 1.

Algorithm 1. K-means initialized by evolutionary computation.

```
1: Randomly initialize a population of individuals, each individuals dimension is $K$.
2: Calculate the fitness function for each individual in the population. First, we parse
   the centroid for each cluster from the sample index based encoding method. Then
   each samples cluster is determined according to the clusters centroids, and the
   WCSS will be used as an individuals fitness function.
3: Apply the evolutionary operations (such as selection, crossover, mutation, etc.)
   of evolutionary algorithms to get the next generation of the population from the
   current population.
4: If the termination condition of evolutionary algorithm meets, get the optimal so-
   lution and go to Step 5, otherwise go to Step 2.
5: Figure out the centroid of each cluster from the optimal individual obtained by
   evolutionary computation.
6: Each samples cluster is determined according to its closest centroid.
7: Calculate the mean vector of all the samples in each cluster. Then use the mean
   vector as the new cluster centroid.
8: If the termination condition of k-means is satisfied, go to Step 9, otherwise go to
   Step 6.
9: Output clustering result.
```

5 Experiments

5.1 Experimental Setup

In this paper, we use six evolutionary computation algorithms for clustering, which are differential evolution (DE) [6], the conventional fireworks algorithm (FWA) [8], enhanced fireworks algorithm (EFWA) [17], evolutionary strategies (ES) [5], genetic algorithms (GA) [4] and particle swarm optimization algorithm (PSO) [7].

For fireworks algorithm and enhanced fireworks algorithm, we use the default parameters from their original papers. We use the java library jMetal [18] to implement the other four algorithms, and the default parameters of jMetal is used for these algorithms. The maximum number of evaluations of all algorithms are set to 25000.

These algorithms will use a lot of random numbers at running time, so the results obtained by the algorithm will be different when run repeatedly. In order to obtain a stable measurement evolutionary computation based clustering, each experiment were run 20 times, and the average of the results will be used as the final result.

Experiments are conducted on eight commonly used document clustering data sets. Stacked auto-encoder is used to extract document feature [19]. At first the tf-idf feature [20] is extracted for the most frequent 2000 words. Then a stacked auto-encoder with the structure of $2000 - 500 - 250 - 125 - 10$ is used to extract abstract document feature. After such feature extraction each document is represented as a 10-dimensional vector. The name of each dataset, the number of samples, the dimension and the number of categories are shown in Table 1.

Table 1. Detailed information of the eight datasets

Dataset	Number of Samples	Dimension	Number of Categories
re0	1504	10	13
re1	1657	10	25
wap	1560	10	20
tr31	927	10	7
tr45	690	10	10
fbis	2463	10	17
la1	3204	10	6
la2	3075	10	6

Clustering algorithms need to set the number of clusters K in advance. These data in the dataset have the original category, we set the number of clusters equal to the number of original categories.

5.2 Comparison among Different Encoding Methods

In this section we will compare cluster centroid based encoding, sample category based encoding and sample index based encoding. The average WCSS over all of the 8 data sets is shown in Table 2.

Table 2. Within-cluster sum of squares of cluster centroid based encoding, sample category based encoding and sample index based encoding

Evolutionary Algorithms	cluster centroid	sample category	sample index
DE	935.08	1582.72	828.44
EFWA	1183.19	1645.08	814.41
ES	920.33	1425.40	800.64
FWA	1023.67	1634.44	819.05
GA	940.97	1524.56	811.12
PSO	1216.97	1639.51	881.40

We can see from the above results that the proposed sample index based encoding performs better than the two existing ways of encoding after optimization by all of the six algorithms. This shows that the proposed encoding method can effectively reflect the essence of the document sets structure, which made the optimization easily to be done. Thus the proposed sample index based encoding method has a great potential in the future development of clustering.

From Table 2, we can get the performance of different evolutionary computation algorithms. Evolutionary strategy (ES) performed best among all of the optimization algorithms, which was followed by the genetic algorithm (GA), enhanced fireworks algorithm (EFWA), fireworks algorithm (FWA) and differential evolution (DE), the worst one is the particle swarm optimization (PSO). Therefore the evolutionary computation algorithm used also has an import impact on the clustering performance.

We also compare the clustering results between evolutionary computation based clustering and k-means. The average WCSS of k-means over the 8 data sets is 811.32. As shown in Table 2, evolutionary strategy achieves better result than the k-means, so it can be used as a new and effective clustering methods. However, several other optimization algorithms effect is not significant. While the usage of evolutionary computation to initialize k-means can effectively improve the quality of clustering, experimental results are shown in the next section.

5.3 K-means Initialized by Evolutionary Computation

Fig. 1 gives the average WCSS over eight datasets of three clustering algorithms. The three clustering algorithms include clustering using evolutionary computation directly, k-means and k-means initialized by evolutionary computation. The average WCSS of clustering using PSO directly is 881.4, while other clustering algorithms are all below 835. We cut this extreme value in Fig. 1 to get a suitable figure; only the part below 835 is shown in Fig. 1.

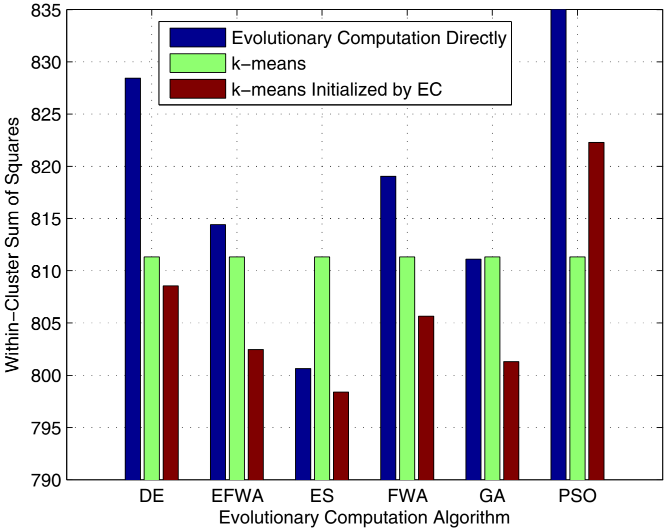

Fig. 1. Within-cluster sum of squares of clustering using evolutionary computation directly, k-means and k-means initialized by evolutionary computation

From the figure we can see that k-means initialized by differential evolution, enhanced fireworks algorithm, evolution strategy, fireworks algorithm and genetic algorithm is superior to original k-means and the direct evolutionary computation algorithms. Therefore the initialization strategy of k-means is able to improve the performance of clustering evidently.

It is easy to fall into local optimum for traditional k-means initialized at random. While k-means initialized by evolutionary computation will locate the initial centroids near the optimal centroids. In such case k-means who has excellent local searching ability will find the optimal centroids easily. The clustering performance of particle swarm optimization is slightly poor. This is because particle swarm optimization doesn't converge well for clustering.

6 Conclusions

This paper introduces the basic principle and procedures of k-means and clustering using evolutionary computation. A novel encoding method based on sample index is proposed in this paper. We combine k-means and evolutionary computation by initializing k-means by evolutionary computation to enhance the clustering performance. At last this paper gives the experimental results of evolutionary computation based clustering over six common evolutionary computation algorithms.

The proposed sample index based encoding method significantly outperforms cluster centroid based encoding and sample category based encoding. The encoding method based on sample index is able to restrict the centroids within the training data. The evolutionary computation algorithms will concentrate on meaningful search space to get a better solution. Whats more, the search space of this encoding method is much smaller than the other two encoding methods due to its lower dimension, therefore evolutionary computation algorithms will find the optimal centroids more easily.

K-means initialized by evolutionary computation is superior to the original k-means and using evolutionary computation directly. Evolutionary computation is good at global search, while k-means is good at local search. Initializing k-means by evolutionary computation is able to combine the two advantages and improve the clustering performance.

Acknowledgments. This work was supported by National Natural Science Foundation of China (NSFC) with Grant No. 61375119, No. 61170057 and No. 60875080.

References

1. Hartigan, J.A., Wong, M.A.: Algorithm as 136: A k-means clustering algorithm. Applied Statistics, 100–108 (1979)
2. Johnson, S.C.: Hierarchical clustering schemes. Psychometrika 32(3), 241–254 (1967)
3. Moon, T.K.: The expectation-maximization algorithm. IEEE Signal Processing Magazine 13(6), 47–60 (1996)

4. Goldberg, D.E., et al.: Genetic algorithms in search, optimization, and machine learning, vol. 412. Addison-wesley, Reading (1989)
5. Liem, K.F.: Evolutionary strategies and morphological innovations: cichlid pharyngeal jaws. Systematic Biology 22(4), 425–441 (1973)
6. Storn, R., Price, K.: Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4), 341–359 (1997)
7. Kennedy, J., Eberhart, R., et al.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, vol. 4, pp. 1942–1948 (1995)
8. Tan, Y., Zhu, Y.: Fireworks algorithm for optimization. In: Tan, Y., Shi, Y., Tan, K.C. (eds.) ICSI 2010, Part I. LNCS, vol. 6145, pp. 355–364. Springer, Heidelberg (2010)
9. Maulik, U., Bandyopadhyay, S.: Genetic algorithm-based clustering technique. Pattern Recognition 33(9), 1455–1465 (2000)
10. Bandyopadhyay, S., Maulik, U.: An evolutionary technique based on k-means algorithm for optimal clustering in rn. Information Sciences 146(1), 221–237 (2002)
11. Das, S., Abraham, A., Konar, A.: Automatic clustering using an improved differential evolution algorithm. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 38(1), 218–237 (2008)
12. Paterlini, S., Krink, T.: Differential evolution and particle swarm optimisation in partitional clustering. Computational Statistics & Data Analysis 50(5), 1220–1247 (2006)
13. Van der Merwe, D., Engelbrecht, A.P.: Data clustering using particle swarm optimization. In: The 2003 Congress on Evolutionary Computation, CEC 2003, vol. 1, pp. 215–220. IEEE (2003)
14. Chen, C.Y., Ye, F.: Particle swarm optimization algorithm and its application to clustering analysis. In: 2004 IEEE International Conference on Networking, Sensing and Control, vol. 2, pp. 789–794. IEEE (2004)
15. Forsati, R., Mahdavi, M., Shamsfard, M., Reza Meybodi, M.: Efficient stochastic algorithms for document clustering. Information Sciences 220, 269–291 (2013)
16. Ahmadyfard, A., Modares, H.: Combining pso and k-means to enhance data clustering. In: International Symposium on Telecommunications, IST 2008, pp. 688–691. IEEE (2008)
17. Zheng, S., Janecek, A., Tan, Y.: Enhanced fireworks algorithm. In: 2013 IEEE Congress on Evolutionary Computation (CEC), pp. 2069–2077. IEEE (2013)
18. Durillo, J.J., Nebro, A.J., Alba, E.: The jmetal framework for multi-objective optimization: Design and architecture. In: 2010 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8. IEEE (2010)
19. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science 313(5786), 504–507 (2006)
20. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc., New York (1986)