Approach to the construction of the genetic fuzzy clustering algorithm

Panevskiy V.V.[1], Polykov V.M.[2], Buckanov D.M.[2]

[1] *Cloud services, Mirantis. Inc., Lenina ave., Kharkiv, Ukraine*
[2] *Belgorod goverment university by Shuckov V.G.,46 Kostukova str., Belgorod, Russia*

amangions@gmail.com

Abstract. *When you work with a lot of data often face the task of class data mining. One of the common problems is the clustering of data. Of particular interest is the fuzzy clustering, providing more complete information about the structure of data clusters. Offers an improved fuzzy clustering algorithm, which can handle local extremes of the convergence function of the algorithm by integrating the fuzzy c-means algorithm with a genetic algorithm. The composition of the genetic algorithm and the algorithm of a fuzzy c-means clustering has also improved characteristics convergence compared to conventional combinatorial search. Thus, we propose an approach that combines both algorithms that complement each other and compensate for lack of both algorithms separately.*

Keywords

Clustering, Genetic algorithms, Fuzzy logic.

1 Introduction

In most distributed systems, nodes constituting the system can be arranged in different territorial areas. This allows you to get the system is resistant to various global factors. Although in reality the access speed to the same network may differ from the use of various network devices and software. And the developer seeks to optimize the operation of a distributed system with the speed networking.

Consider the computer network of n nodes, in which the speed of access between nodes can be represented as a weighted graph, where the vertices of the graph correspond to the nodes and edges represent the logical connection between the nodes. Weight of the arc determines the response time between nodes. The task to break this network into K clusters. It is necessary for the synchronization algorithm in the network computation. Each cluster is constructed minimum spanning tree, and then checked that each node of the tree has been synchronized (located at a certain point.) After that, the algorithm is run for all the tops of the trees in the network. Thus, to synchronize the network. Clustering access speed between network nodes will reduce the costs of the transmission of information, ie minimize the weighted sum of all distances, thereby ensuring rapid exchange of messages between nodes within the selected cluster.

After reviewing the existing clustering approaches, the following conclusions:

• Genetic algorithms and artificial neural networks are well parallelized.
• Genetic algorithms and tempering method is carried out a global search, but the tempering method converges very slowly.
• k-Means works fast and simple to implement, but only creates clusters, similar to the hypersphere.
• Hierarchical algorithms yield optimal partition into clusters, but the complexity of quadratic.
• In practice, the best proven hybrid approaches, where the grinding is performed by the cluster k-Means, and the original partition - one of the more powerful methods.

2 Fuzzy clustering algorithm

It is proposed fuzzy clustering for more information about the system and to modify the structure of the clusters in the case without the need to run the algorithm. A search for the optimal solutions is proposed to use a genetic algorithm [1] [2].

The basis of the hybrid algorithm put the well-known fuzzy clustering algorithm

C-means [3], we consider it in more detail. This algorithm calculates for each object xi degree of u_{ik} supplies to each of the k clusters.

- Step 1

Initialization:
• N - number of vectors or points
• K - number of points clusters
• E - the level of precision
• Measure the distance as the Euclidean distance
• The index of fuzziness q = 1.5
• Matrix accessories u, determines whether the vectors x_j to the centers of the clusters c_k

- Step 2

Adjustment of the position of cluster centers c_k.

$$c_k = \frac{\sum_{k=1}^{K}(u_{jk})^q * x_j}{\sum_{j=1}^{N}(u_{jk})^q}$$

- Step 3

Adjustments values of accessories u_{jk}.

$$u_{ik} = \frac{1/||x_i - c_k||^{1/(1-q)}}{\sum_{k=1}^{K}(1/||x_i - c_k||^{1/(q-1)})}$$

- Step 4. Stop the algorithm.

Fuzzy clustering algorithm stops when the following conditions:

$$\| U (t+1) – U(t) \| <= E$$

E – above a predetermined level of accuracy.

In the context of developing hybrid algorithm of C-means algorithm is used only steps 2 and 3. The disadvantages of the algorithm C-means:

1) The algorithm can be completed on a local extremum.
2) It is sensitive to the initialization.

The impact of these deficiencies can be substantially reduced by developing a hybrid version of the algorithm. A genetic algorithm, in principle, is addressing these shortcomings, but raises a new question of the convergence of the algorithm. Improve the performance of the convergence of the genetic algorithm can be achieved by combining it with the algorithm c-means. That is, global search will be carried out with alternating steps of gradient descent for faster convergence. c-means algorithm with greatly depends on initialization (and accuracy of the initial cluster centers random distribution), and finally we cannot obtain optimal partitioning. The genetic algorithm with a relatively large number of initial populations eliminates this problem.

3 The hybrid algorithm based on fuzzy clustering algorithm c-means

Clustering problem can be solved by using a genetic algorithm, it is necessary to select the chromosome. Chromosome will be the matrix U accessory clusters of points, the matrix will be written as a string s, so it will be easier to perform the genetic operations.

Fitness function will be:

$$S(U) = \sum_{k=1}^{K}\sum_{i=1}^{N}(u_{ki})^q \left||x_i - c_k\right|| \rightarrow min$$

x_i - clustered data elements i = 1 .. N
c_k - the centers of clusters k = 1 .. K
u_{ki} - belonging i-th point of the k-th cluster

Need to find a matrix accessories U *, which minimizes

$$S(U^*) = \min\{S(U)\}$$

Basic notation and algorithm steps:

• K - number of clusters
• N - number of points (nodes)
• Pc - the likelihood of a crossover on a pair of chromosomes
• Pm - the likelihood of a mutation

1) Initiate.
The population P is generated at random, but with the restrictions on empty clusters. With a random distribution of points in clusters, number of points for each cluster can be no more than p (the maximum integer less than N / K). Thus avoids empty clusters to initialize the initial population.

2) Selection.
The selection operator randomly selects chromosomes from the previous population according to the following distribution:

$$P(s_i) = \frac{F(s_i)}{\sum_{j=1}^{N} F(s_j)}$$

For an arbitrary choice of using the strategy of "Roulette". For the next population of the fittest individuals are selected. With such a selection of members of the population with higher fitness are chosen more often than individuals with low. In this context, a fitness function value for the selected chromosome su depends on S (U). In order to optimize the S (U), the solution string with a relatively small quadratic error must have a relatively high fitness function value [1].

3) Crossover.
Genetic algorithm executes a crossover for n selected individuals with a given probability Pc, for which the n rows are divided by n / 2 randomly. For each pair applies crossover with probability Pc. The expression 1 - Pc denotes the probability of preservation of individuals who are moving to the stage of the mutation. Progeny obtained by replacing the crossover of their parents and will also go to the mutation. Single-point crossover works as follows. First, randomly select one of the l - 1 break points (l - number of genes). Parent structure at this point is torn into two segments. The respective segments of different parents are glued and genotype results in two offspring.

4) The mutation.
Mutation selected value of alleles depending on the distance between the center point of the cluster. The mutation operator is probabilistic. For the mutations most likely choice with the number of alleles of the cluster closest to the data point.

Let $d_j = d(x_i, c_j)$ - Euclidean distance between the point xi and the center c_j. The value of the alleles is replaced by the following value with the following distribution:

$$p_j = \frac{c_m d_{max} - d_j}{\sum_{j=1}^{k}(c_m d_{max} - d_j)}$$

cm - a constant, typically > = 1 and $d_{max} = \max\{dj\}$

It is worth paying attention to the one-point clusters. In the case of a singleton cluster $d_j = 0$ (It could also be, if in a cluster multipoint data point coincides with the center). The mutation occurs only when $d_j > 0$ with the chosen probability Pm.

5) Perform step of the algorithm C-means.
This step serves to improve the convergence of the algorithm, it is to perform one iteration of the C-means algorithm for each member of the population. In other words, we get the additional mutation of chromosomes, which refines the solution.

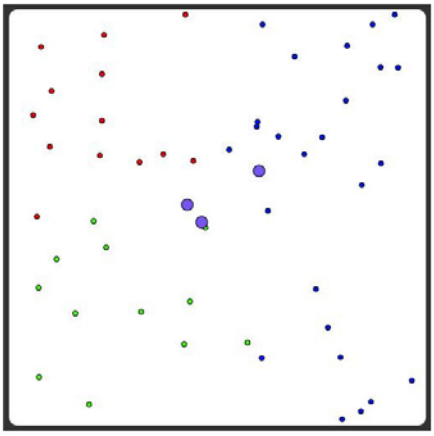

Fig. 1. The visual presentation of the results of the algorithm.

Figure 1 is a partition of a set of points in the plane of the 3 clusters (blue, green, red). This may in fact be some network model, where RTT is the distance between the nodes between them. Large purple dots are centers of the corresponding clusters, which ones are nodes too.

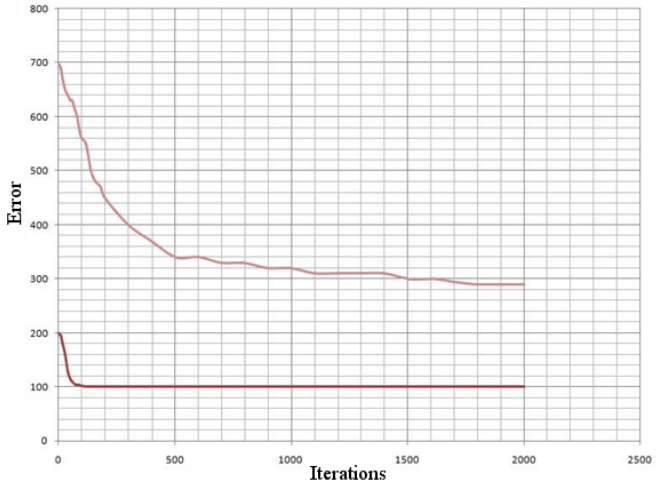

Fig. 2. Graph of the convergence of the algorithm and a hybrid of classical fuzzy clustering algorithm.

On figure 2 shows that the hybrid algorithm converges faster and with better results than the algorithm c-means. The initial data set for clustering a set of points in the plane was, the distance between which is consistent with the response time between nodes. If the result-means algorithm on the same set of test data gave similar results, the results of genetic algorithm were different, but still with less error than the classical algorithm [4-6].

Hybrid algorithm used in a distributed system. To experiment, was generated a random set of points in the plane of dimension 1500. Distance between the points corresponding to the time response between the nodes. Graph algorithms convergence given me in the article complies average of 20-30 test runs. The genetic algorithm performed 3000 epochs. Tests were conducted on a laptop with a core i7 processor and 8 gb of RAM.

The basic procedure of the genetic algorithm:

```
Input:
   Mutation probability, Pm;
   The population size, N;
   The maximum number of evolutions, MAX_GEN;
Output: a string solution (chromosome), s *;

GeneticC-Means() {
        Initialize the populations of P;
         geno = MAX_GEN ;
         s* = P1; (Pi - is the i-th row of the matrix P)
         while (geno > 0)
                Calculate the fitness of each row of the matrix P;
                  P* = Selection (P);
                  for i = 1 to N ,
                          Pi = Mutation(Pi);
                          C-Means(Pi);
                   S = row of the matrix P, which is the matrix U has a minimum error S;
                  if (S(U*) > S(U))  s* = s;
    endWhile
}
```

Function drand () returns the number of uniformly distributed on the interval [0, 1]

Mutation:

```
Mutation (sw) {
        for i = 1 to N
                if (drand() < Pm )
                        Calculate the cluster centers cj-th, corresponding to sw
                        for j = 1 to K
                                dj = d(xi,cj);
                        endFor
                        if (dsw(i) > 0)
                                dmax = max{d1, d2, …, dK}
                                for j = 1 to K
                                        pj = (cm*dmax — dj) /
                                 endFor
                                sw (i) = number of randomly chosen from {1, 2, K} according to
                                distribution {p1, p2, ..., pK};
                        endIf
                endIf
        endFor
}
```

References

[1] D. B. Fogel, “An introduction to simulated evolutionary optimization,”
IEEE Trans. Neural Networks, vol. 5, no. 1, pp. 3–14, 1994.
[2] J. H. Holland, Adaptation in Natural and Artificial Systems. Ann
Arbor, MI: Univ. of Michigan Press, 1975.
[3] A. K. Jain and R. C. Dubes, Algorithms for Clustering Data. Englewood Cliffs, NJ: Prentice-Hall, 1989.
[4] R. W. Klein and R. C. Dubes, “Experiments in projection and clustering
by simulated annealing,” Pattern Recognit., vol. 22, pp. 213–220, 1989.
[5] S. Z. Selim and K. Alsultan, “A simulated annealing algorithm for the
clustering problem,” Pattern Recognit., vol. 10, no. 24, pp. 1003–1008,
1991.
[6] Neural networks, genetic algorithms and fuzzy systems: Transl. from pol. Rudinskogo I. D., - M.: Telecom hot line,
2006. – 452 p.: pic.