A Novel Genetic Clustering Algorithm with Variable-length Chromosome Representation

Ming-an Zhang
Institute of Software, Chinese Academy of Sciences
4# South Fourth Street,
Zhongguancun, Beijing 100190,
PR.China
86 010 62661154
171166789@qq.com

Yong Deng*
Institute of Software, Chinese Academy of Sciences
4# South Fourth Street,
Zhongguancun, Beijing 100190,
PR.China
86 010 62661154
dengyong@iscas.ac.cn

Dong-xia Chang
Institute of Information Science,
Beijing jiaotong University
No.3 Shang Yuan Cun,Hai Dian
District Beijing 100040, PR.China
86 010 51684108
dxchang@bjtu.edu.cn

ABSTRACT

The paper proposed a new genetic clustering algorithm with variable-length chromosome representation(GCVCR), which can automatically evolve and find the optimal number of clusters as well as proper cluster centers of the data set. A new clustering criterion based on message passing between data points and the candidate centers described by the chromosome are presented to make the clustering problem more effective. The simulation results show the effectiveness of the proposed algorithm.

Categories and Subject Descriptors

I.4.6 Segmentation: pixel classification

General Terms: Algorithms

Keywords

Clustering; Genetic algorithm; Data segmentation

1. INTRODUCTION

People share their daily activities and opinions on social networking websites, opening the floodgates of information that can be analyzed by marketers as well as consumers. However, low barriers to publication and easy-to-use interactive interfaces have contributed to various information quality problems in the social media. Approaches such as data mining have begun to address these challenges[1]. Clustering techniques have been broadly employed in data mining. Clustering algorithms are essentially local search algorithms, using an iterative climbing technique to find the optimal solution[2]. And the algorithms are also apt to fall into a local optimum and the result is sensitive to initialization[3,4]. The paper proposes a clustering genetic algorithm with variable-length chromosome representation.

2. SIMILARITY MEASURE

We propose a similarity measure firstly. It uses two kinds of message, responsibility and availability, exchanged between data points and the candidate centers. Here, the responsibility and availability between the data set X={x_1,…,x_n} and the candidate centers set C={c_1,…,c_n} are defined. For the candidate center set C, an input preference that candidate centers with larger values of input preference are more likely to be chosen as a center. If a priori, the value can be set by the priori information.

$$IP(k) = -\frac{1}{n}\sum_{i=1}^{n} d^2(x_i, c_k) = -\frac{1}{n}\sum_{i=1}^{n}\left\| x_i - c_k \right\|^2, \qquad k = 1,2,\cdots K_i \tag{1}$$

This is the mean distance between a center and all the data point in the data set. This value will be optimized when c_k is the center. Note that the distance measure here is chosen with the Euclidean norm. *IP*(*k*) represents the suitability of the candidate centers as the real one. The responsibility *r*(*i*,*k*), sent from data point x_i to the candidate center c_k, reflects the evidence for how well-suited c_k is as the center for point x_i, taking into account other potential centers for point x_i. The availability *a*(*i*,*k*), sent from candidate center c_k to point x_i, reflects the evidence for how appropriate it would be for point x_i to choose c_k as its center, taking into account the support from other points that c_k should be an center.

$$r(i,k) = d(i,k) - \max_{k's.t.k' \neq k}\{d(i,k')\}, i = 1,2,\cdots,n; k = 1,2,\cdots,K_i \tag{2}$$

Here, *d*(*i*,*k*) denotes the distance between data point x_i and the candidate cluster center c_k. For each candidate cluster center, a self-attribution is defined as:

$$R(k) = IP(k) - \max_{k's.t.k' \neq k}\{d(c_k, c_{k'})\}, \qquad k = 1,2,\cdots,K_i \tag{3}$$

Whereas the above responsibility update lets all candidate centers compete for ownership of a data point.

$$a(i,k) = \min\left\{0, R(k) + \sum_{i',s.t.i' \notin \{i\}} \max\{0, r(i',k)\}\right\}, i = 1,2,\cdots,n; k = 1,2,\cdots,K_i \tag{4}$$

This availability *a*(*i*,*k*) reflects evidence that point c_k is a center. Then the similarity between the data point and the candidate center is defined by the sum of the responsibility *r* and the availability *a*. The similarities between data point x_i and the candidate centers C = {c_1, . . . ,c_K} are

$$s(i,k) = r(i,k) + a(i,k), \qquad k = 1,2,\cdots,K_i \tag{5}$$

then x_i will be assigned to the cluster with the maximum similarity.

3. GENETIC CLUSTERING ALGORITHM

3.1 Chromosome Representation and Population Initialization

A chromosome representation is needed to describe each individual in the population of interest. Extensive experiments

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.
ACM 978-1-4503-2881-4/14/07.

comparing real-valued and binary showed that the real-valued GA is more efficient in terms of CPU time. Here, a real-valued problem-specific chromosome representation is used. Each chromosome is described by a sequence of M=N×K_i real-valued numbers where N is the dimension of the feature space, K_i is the number of clusters described by the chromosome.

$$\mathrm{v} = [v_{11}, v_{12}, \cdots, v_{1N}, v_{21}, v_{22}, \cdots, v_{2N}, \cdots, v_{K_i1}, v_{K_i2}, \cdots, v_{K_iN}] \qquad (6)$$

Each of N values represents a cluster center. The K_i of individuals may take different values, and change in the course of evolution. Therefore, the representation is of variable-length. An initial population of size P can be randomly generated. For each population, K_i is generated randomly, districted by a lower and an upper bound. Then K_i points are chosen randomly from the data set but on the condition that there are no identical points to form a chromosome, presenting the K_i cluster centers. This process is repeated until P chromosomes are generated.

3.2 Fitness Function

The aim of the fitness function is to find a set of centers for which the within-cluster spread is small. The between-cluster spread is large in some sense, and the number of clusters is moderate. Considering the influence of the number of the centers on the clustering result, we define a penalized cost function as

$$J = \frac{1}{K^2} Tr\{S_W^{-1} S_B\} \qquad (7)$$

where S_W is the within-cluster variation and S_B is the between-cluster variation, K is the number of cluster centers.

3.3 Crossover and Mutation

In the evolutionary process, we use crossover and mutation on the individuals to manipulate the centers to evolve chromosomes into possibly better ones. The crossover and mutation operators are selected by a probability. Simple crossover is used here. Note that although the crossover points can fall in different locations in the two individuals, they are districted to fall on the same location within each cluster description. There are 6 mutator operators: perturb, insert, delete, merge, split, and move.

3.4 Algorithm Procedure

(1)Initialize a group of cluster centers with size of P, only nontrivial clustering are considered. Each data point of the set is assigned to the cluster with the new similarity measure. (2)Evaluate each chromosome and copy the best chromosome say p_{best} of the initial population in a separate location. (3)If the termination condition is not reached, go to Step 4. Otherwise, select the best individual from the population as the best clustering result. (4)Select individuals from the population for crossover and mutation. (5)Apply crossover operator to the selected individuals based on the crossover probability. (6)Apply mutation operator to the selected individuals based on the mutation probability. (7)Evaluate the newly generated candidates. (8)Compare the worst chromosome in the new population with p_{best} in term of their fitness values. If the worst one is worse than p_{best}, then replace it by p_{best}. (9)Find the best chromosome in the new population and replace p_{best}. (10)Go back to Step 3.

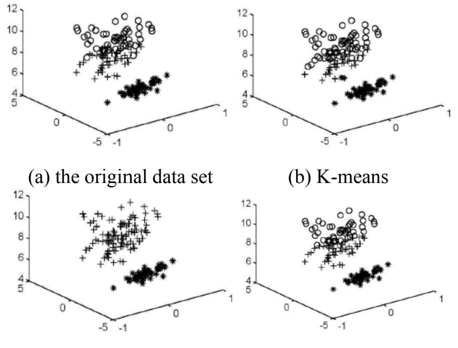
4. SIMULATION

The performances of the GCVCR, GCUK[3] and K-means algorithms[4] are compared through the experiments. The artificial data set used is similar to those in Ref.3, consisting of 150 3D data points distributed over 3 clusters as shown in Fig. 1(a). Each cluster consists of 50 data points. Two of the clusters are overlapping. The population size is taken as 50. The crossover and mutation probabilities for GUCK are pc=0.8 and pm=0.001. For GCVCR, the mutation probability varies follow a inverse proportion function. We have started with a mutation probability value of pm = 0.5. The value is then varied as a step function of the number of iterations until it reaches a value of 0.001. The minimum value of the mutation probability is taken to be 0.001. The probabilities for the mutation operators are set to be equal to 1/6. The total number of generations is equal to 50. For the K-means algorithm, the actual number of clusters is known prior. All the experiments run for 20 independent times. AC is the true number of the clusters. From Tab.1 GCVCR obtains the exact K.

Table 1. The mean and variance of the adjusted rand index

AC	GUCK	GCVCR
3	2.05 ± 0.0500,1	2.09 ± 0.0947,18

Fig.1 provides the clustering results. It is clear that GCVCR is better than other algorithms. K-means results in a significant misclassification from Fig.1(b). GCUK only gets 2 clusters from Fig.1(c). The 2 aliasing clusters are classified to one. GCVCR obtains the correct number of clusters.



(a) the original data set (b) K-means

(C) GCUK (d) GCVCR

Figure 1. The original data set and the clustering results.

5. CONCLUSIONS

This paper presents a new genetic clustering algorithm, which can find automatically the optimal number of clusters as well as the cluster centers in the evolutionary process. Simulation results show that the algorithm proposed has better performance.

6. ACKNOWLEDGMENTS

This work was supported by Beijing Municipal Science Foundation No. 4133092 and 863 program (No.2012AA011206).

7. REFERENCES

[1] Agarwal Nitin, Yiliyasi Yusuf. Information quality challenges in social media. Proceedings of the 2010 International Conference on Information Quality, 2010.

[2] Chang D X, Zhang X D. A robust dynamic niching genetic algorithm with niche migration for automatic clustering problem. Pattern Recognition, 2010, 43(7): 1346-1360.

[3] Bandyopadhyay S, Maulik U, Genetic clustering for automatic evolution of clusters and application to image classification. Pattern Recognition, 2002, 35(6): 1197-1208.

[4] Kanungo T, Mount D, et al. An efficient K-means clustering algorithm: Analysis and implementation. IEEE Trans. Pattern Anal. Mach. Inteli., 2002, 24(7): 881-892.