Genetic algorithm-based clustering technique

Ujjwal Maulik, Sanghamitra Bandyopadhyay
Presented by Hu Shu-chiung
2004.05.27

References:

1. Genetic algorithm-based clustering technique
2. Slide of Genetic Algorithms, present by St. Chen 2004
3. Slide of Cluster Analysis, Berlin Chen 2004

Outline

- Introduction
- Clustering—K-means algorithm
- Clustering using genetic algorithms
- Implementation results
- Discussion and Conclusion

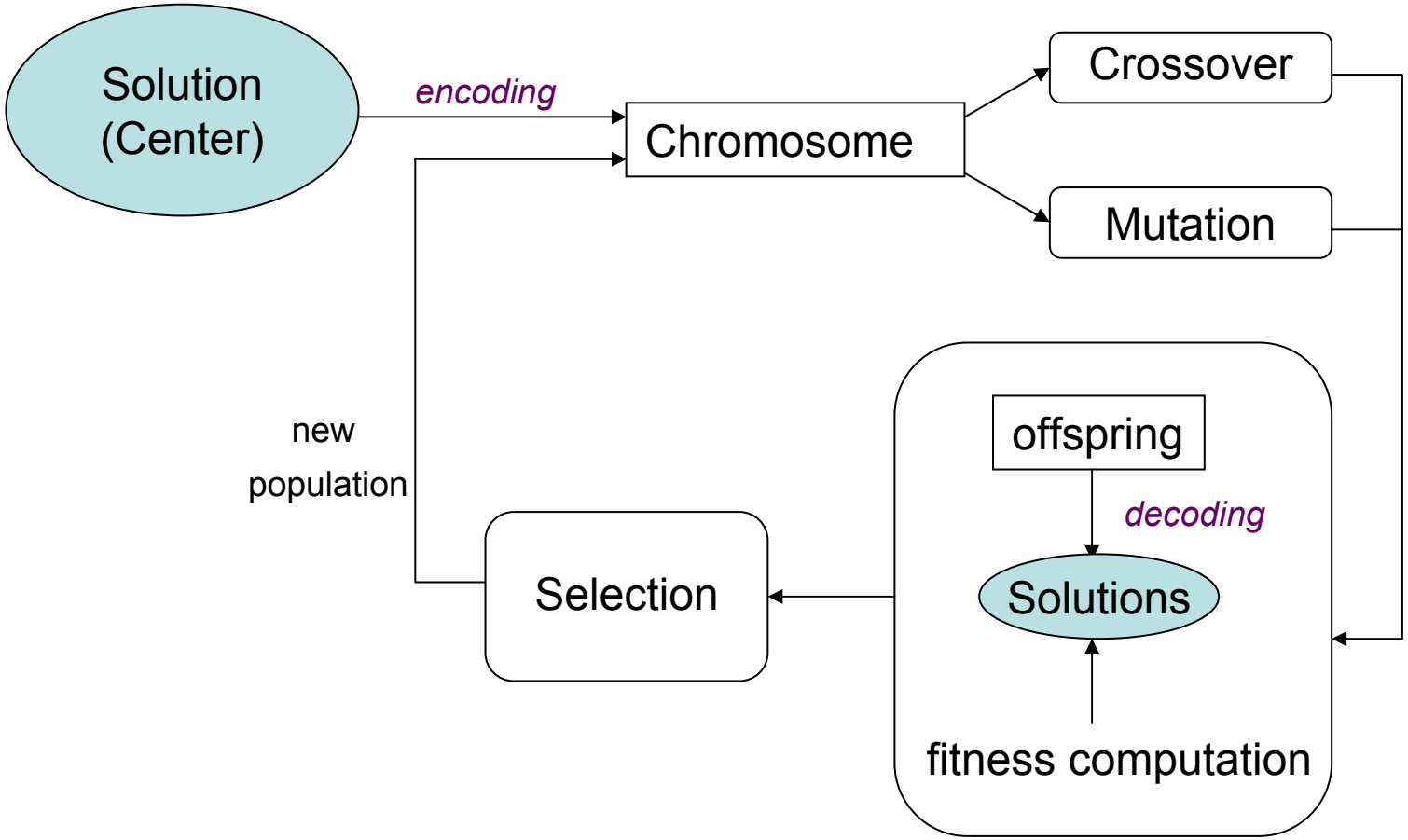
Introduction

Genetic Algorithm

- Genetic algorithms (GAs) are randomized search and optimization techniques guided by the concepts of natural selection and evolutionary processes.

Introduction

The general Structure of GAs



Introduction

Clustering

- Unsupervised classification
  - classification of unlabeled data
- Clustering
  - an important unsupervised classification technique where a set of pattern, usually vectors in a multi-dimensional space, are grouped into clusters in such a way that patterns in the same cluster are similar in some sense and patterns in different clusters are dissimilar in the same sense.
- First define a measure of similarity which will establish a rule for assigning patterns to the domain of a particular cluster center.
  - One such measure of similarity may be Euclidean Distance D between two patterns x and z defined by

    $D = \| x - z \|$. Smaller the distance, greater is the similarity.

Introduction

Clustering

- Clustering in *N*-dimensional Euclidean space R^N is the process of partitioning a given set of *n* points into a number, say K, of groups (or, clusters) based on some similarity / dissimilarity metric. Let the set of *n* points $\{x_1, x_2, \ldots, x_n\}$ be represented by the set *S* and the K clusters be represented by $C_1, C_2, \ldots, C_K$. Then

$$C_i \neq \emptyset \quad \text{for } i = 1, \ldots, K,$$

$$C_i \cap C_j = \emptyset \quad \text{for } i = 1, \ldots, K, j = 1, \ldots, K \text{ and } i \neq j$$

$$\text{and } \bigcup_{i=1}^{K} C_i = S.$$

Introduction

K-means Algorithm

- Step 1: Choose K initial cluster centers $z_1, z_2, \ldots, z_K$, randomly from the n points $\{x_1, x_2, \ldots, x_n\}$.
- Step 2: Assign point $x_i, i = 1,2,\ldots,n$ to cluster C_j, $j \in \{1,2,\ldots,K\}$ iff
  $\| x_i - z_j \| < \| x_i - z_p \|$, $p = 1,2,\ldots,K$, and $j \neq p$.
- Step 3: Compute new cluster centers $z_1, z_2, \ldots, z_K$ as follows:

$$\mathbf{z}_i^* = \frac{1}{n_i} \sum_{x_j \in C_i} \mathbf{x}_j, \quad i = 1, 2, \ldots, K,$$

  where n_i is the number of elements belonging to cluster C_j
- Step 4: If $z_i = z_i$, $i = 1,2,\ldots,K$ then terminate. Otherwise continue from step 2

Note: In case the process does not terminate at Step 4 normally, then it is executed for a maximum fixed number of iterations.

Clustering using GA

- Basic principle

```
Begin
1.  t=0
2.  initialize population P(t)
3.  compute fitness P(t)
4.  t = t+1
5.  if termination criterion achieved go to step 10
6.  select P(t) from P(t-1)
7.  crossover P(t)
8.  mutate P(t)
9.  go to step 3
10.  Output best and stop
End
```

Basic steps in GAs.

Clustering using GA

- Euclidean distances of the points from their respective cluster centers. Mathematically, the clustering metric μ for the K clusters C_1, C_2,…, C_K is given by

$$\mathcal{M}(C_1, C_2, \ldots, C_K) = \sum_{i=1}^{K} \sum_{x_j \in C_i} \|\mathbf{x}_j - \mathbf{z}_i\|.$$

- The task of the GA is to search for the appropriate cluster centers z_1, z_2,…, z_K such that the clustering metric μ is minimized.

GA-clustering algorithm

- String representation
  - Each string is a sequence of real numbers representing the K cluster centers. (Float-point representation)
  - An N-dimensional space, the length of a chromosome is N*K words,

➔ ($\underline{N_1}$ $\underline{N_2}$ … $\underline{N_K}$) ,

Center of C_1 **Center of C_2** **Center of C_K**

for each N_i is a N-dimensional space number, i = 1,2,…K

Example 1

Example Let $N = 2$ and $K = 3$, i.e., the space is two-dimensional and the number of clusters being considered is three. Then the chromosome

51.6 72.3 18.3 15.7 29.1 32.2

represents the three cluster centres (51.6, 72.3), (18.3, 15.7) and (29.1, 32.2). Note that each real number in the chromosome is an indivisible gene.

GA-clustering algorithm

- Population initialization
  - The K cluster centers encoded in each chromosome are initialized to K randomly chosen points from the data set. This process if repeated for each of the P chromosomes in the population, where P is the size of the population.

GA-clustering algorithm

- Fitness computation (Two phases)
  - In the first phase, the clusters are formed according to the centers encoded in the chromosome under consideration.
  - Assign each point x_i, i=1,2,…,n, to one of the clusters C_j with center zj such that

$$\|\mathbf{x}_i - \mathbf{z}_j\| < \|\mathbf{x}_i - \mathbf{z}_p\|, p = 1, 2, \ldots, K, \text{ and } p \neq j.$$

  - After the clustering is done, the cluster centers encoded in the chromosome are replaced by the mean points of the respective clusters. For cluster C_i, the new center z_i is computed as

$$\mathbf{z}_i^* = \frac{1}{n_i} \sum_{\mathbf{x}_j \in C_i} \mathbf{x}_j, \quad i = 1, 2, \ldots, K.$$

Example 2

Example 2. The first cluster centre in the chromosome considered in Example 1 is (51.6, 72.3). With (51.6, 72.3) as centre, let the resulting cluster contain two more points, viz., (50.0, 70.0) and (52.0, 74.0) besides itself i.e., (51.6, 72.3). Hence the newly computed cluster centre becomes ((50.0 + 52.0 + 51.6)/3, (70.0 + 74.0 + 72.3)/3) = (51.2, 72.1). The new cluster centre (51.2, 72.1) now replaces the previous value of (51.6, 72.3).

GA-clustering algorithm

- Fitness computation (Two phases)
  - Subsequently, the clustering metric μ is computed as follows:

$$\mathcal{M} = \sum_{i=1}^{K} \mathcal{M}_i,$$

$$\mathcal{M}_i = \sum_{\mathbf{x}_j \in C_i} \|\mathbf{x}_j - \mathbf{z}_i\|.$$

  - The fitness function is defined as $f = 1/\mu$.

Maximization of the fitness function leads to minimization of μ

GA-clustering algorithm

- Selection
  - In this article, a chromosome is assigned a number of copies, which is proportional to its fitness in the population, that go into the mating pool for further genetic operations. Roulette wheel selection is one common technique that implements the proportional selection strategy.
- Crossover
  - A probabilistic process that exchanges information between two parent chromosomes for generating two child chromosomes. In this article, single point crossover with a fixed crossover probability of μ_c is used.

GA-clustering algorithm

- Mutation
  - Each chromosome undergoes mutation with a fixed probability μ_m. Floating point representation (chromosomes) are used in this article, we use following mutation. A number δ in the range [0,1] is generated with uniform distribution. If the value at a gene position is v, after mutation it becomes

$$v \pm 2 * \delta * v, \quad v \neq 0,$$

$$v \pm 2 * \delta, \quad v = 0.$$

The '+' or '-' sign occurs with equal probability. we could have implemented mutation as

$$v \pm \delta * v.$$

GA-clustering algorithm

- Termination criterion
  - In this article the processes of fitness computation, selection, crossover, and mutation are executed for a maximum number of iterations.
  - Thus on termination, this location contains the centers of the final clusters.

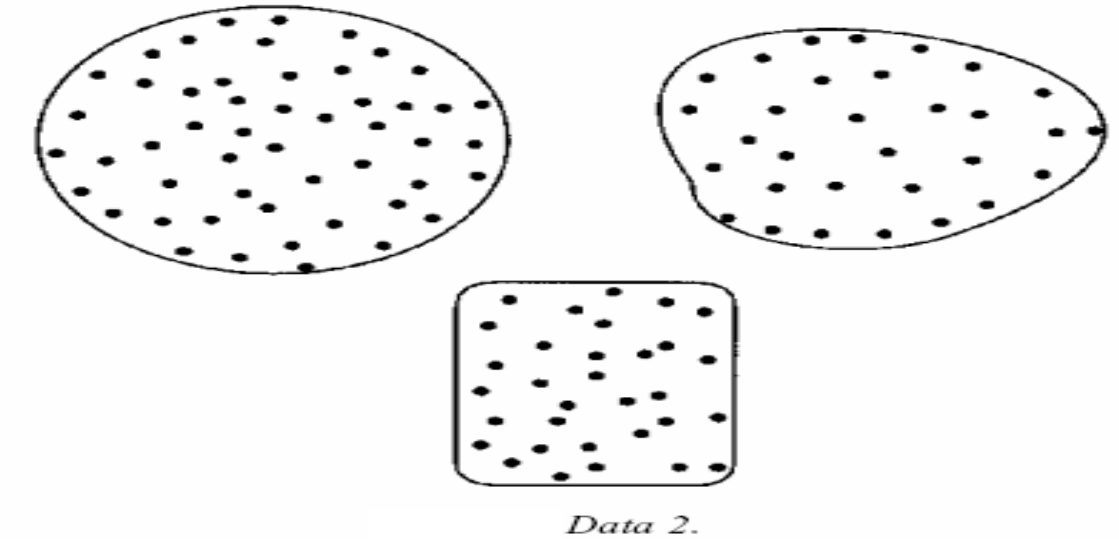
Implementation results

- The experimental results comparing the GA-clustering algorithm with the K-means algorithm are provided for four artificial data sets and three real-life data sets, respectively.

Implementation results

Artificial data sets

- *Data 1*: This is a non-overlapping two-dimensional dataset where the number of clusters is two. It has 10 points. The value of *K* is chosen to be 2 for this data set.
- *Data 2*: This is a non-overlapping two-dimensional dataset where the number of clusters is three. It has 76 points. The value of *K* is chosen to be 3 for this data set.



Data 2.

Implementation results

Artificial data sets

- *Data 3*: This is an overlapping two-dimensional triangular distribution of data points having nine classes where all the classes are assumed to have equal a priori probabilities (= 1/9). It has 900 data points. The *X*-*Y* ranges for the nine classes are as follows:

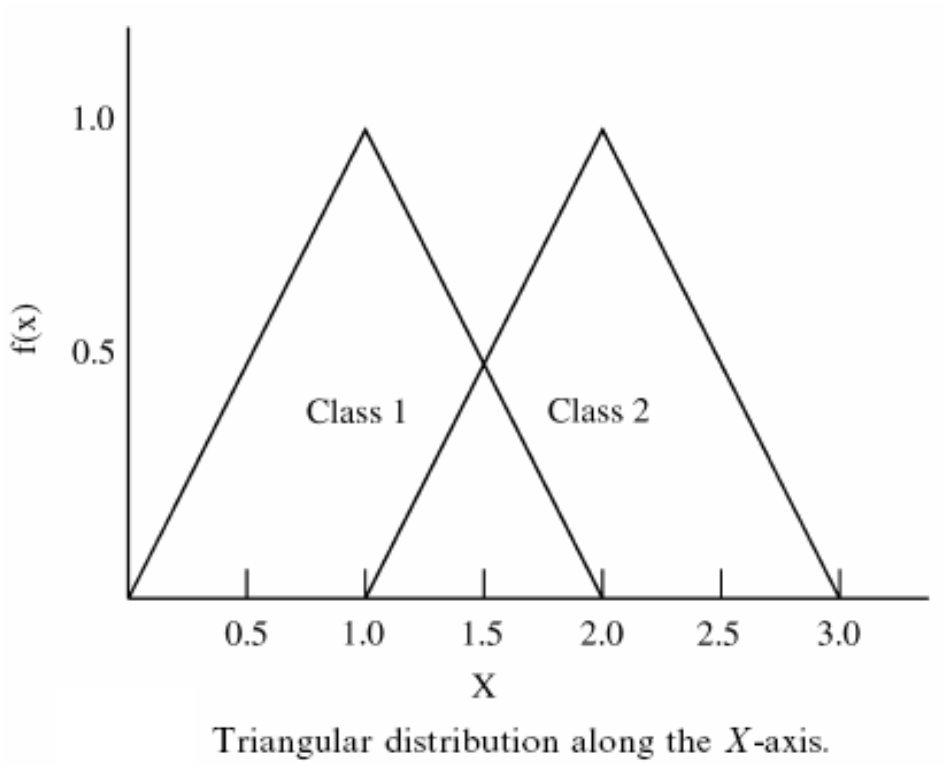
Class 1: $[-3.3, -0.7] \times [0.7, 3.3]$,
Class 2: $[-1.3, 1.3] \times [0.7, 3.3]$,
Class 3: $[0.7, 3.3] \times [0.7, 3.3]$,
Class 4: $[-3.3, -0.7] \times [-1.3, 1.3]$,
Class 5: $[-1.3, 1.3] \times [-1.3, 1.3]$,
Class 6: $[0.7, 3.3] \times [-1.3, 1.3]$,
Class 7: $[-3.3, -0.7] \times [-3.3, -0.7]$,
Class 8: $[-1.3, 1.3] \times [-3.3, -0.7]$,
Class 9: $[0.7, 3.3] \times [-3.3, -0.7]$.

- *Data 4*: This is an overlapping ten-dimensional data set generated using a triangular distribution of the form, two classes. It has 1000 data points.

Implementation results

Artificial data sets

- Thus the domain for the triangular distribution for each class and for each axis is 2.6.



Triangular distribution along the X-axis.

Implementation results

Real-life data sets

- Vowel *data*:
  - This data consists of 871 Indian Telugu vowel sounds. The data set has three features $F1$, $F2$ and $F3$, corresponding to the first, second and third vowel formant frequencies, and six overlapping classes $\{\delta, a, i, u, e, o\}$. The value of K is therefore chosen to be 6 for this data.
- Iris data:
  - This data represents different categories of irises having four feature values. The four feature values represent the sepal length, sepal width, petal length and the petal width in centimeters. It has three classes (with some overlap between classes 2 and 3) with 50 samples per class. The value of K is therefore chosen to be 3 for this data.
- Crude oil data:
  - This overlapping data has 56 data points, 5 features and 3 classes. Hence the value of K is chosen to be 3 for this data set.

Implementation results

- GA-clustering is implemented with the following parameters:
  - μ_c =0.8, μ_m = 0.001,
  - The population size P is taken to be 10 for Data 1, 100 for others.
- For K-means algorithm, a fixed maximum of 1000 iterations in case.

Implementation results

Table 1
$\mathcal{M}$ obtained by K-means algorithm for five different initial configurations for *Data 1* when $K = 2$

Initial configuration	K-means
1	5.383132
2	2.225498
3	2.225498
4	5.383132
5	2.225498

Table 2
$\mathcal{M}$ obtained by GA-clustering algorithm for five different initial populations for *Data 1* after 100 iterations when $K = 2$

Initial population	GA-clustering
1	2.225498
2	2.225498
3	2.225498
4	2.225498
5	2.225498

Table 3
$\mathcal{M}$ obtained by K-means algorithm for five different initial configurations for *Data 2* when $K = 3$

Initial configuration	K-means
1	51.013294
2	64.646739
3	67.166768
4	51.013294
5	64.725676

Table 4
$\mathcal{M}$ obtained by GA-clustering algorithm for five different initial populations for *Data 2* after 100 iterations when $K = 3$

Initial population	GA-clustering
1	51.013294
2	51.013294
3	51.013294
4	51.013294
5	51.013294

Implementation results

Table 5
M obtained by K-means algorithm for five different initial configurations for *Data 3* when $K = 9$

Initial configuration	K-means
1	976.235607
2	976.378990
3	976.378990
4	976.564189
5	976.378990

Table 6
M obtained by GA-clustering algorithm for five different initial populations for *Data 3* after 100 iterations when $K = 9$

Initial population	GA-clustering
1	966.350481
2	966.381601
3	966.350485
4	966.312576
5	966.354085

Table 7
M obtained by K-means algorithm for five different initial configurations for *Data 4* when $K = 2$

Initial configuration	K-means
1	1246.239153
2	1246.239153
3	1246.236680
4	1246.239153
5	1246.237127

Table 8
M obtained by GA-clustering algorithm for five different initial populations for *Data 4* after 100 iterations when $K = 2$

Initial population	GA-clustering
1	1246.221381
2	1246.218355
3	1246.218355
4	1246.218355
5	1246.218355

Implementation results

Table 9
$\mathcal{M}$ obtained by K-means algorithm for five different initial configurations for *Vowel* when $K = 6$

Initial configuration	K-means
1	157460.164831
2	149394.803983
3	161094.118096
4	149373.097180
5	151605.600107

Table 10
$\mathcal{M}$ obtained by GA-clustering algorithm for five different initial populations for *Vowel* after 100 iterations when $K = 6$

Initial population	GA-clustering
1	149346.490128
2	149406.851288
3	149346.152189
4	149355.823103
5	149362.780998

Table 11
$\mathcal{M}$ obtained by K-means algorithm for five different initial configurations for *Iris* when $K = 3$

Initial configuration	K-means
1	97.224869
2	97.204574
3	122.946353
4	124.022373
5	97.204574

Table 12
$\mathcal{M}$ obtained by GA-clustering algorithm for five different initial populations for *Iris* after 100 iterations when $K = 3$

Initial population	GA-clustering
1	97.10077
2	97.10077
3	97.10077
4	97.10077
5	97.10077

Implementation results

Table 13
M obtained by K-means algorithm for five different initial configurations for *Crude Oil* when $K = 3$

Initial configuration	K-means
1	279.743216
2	279.743216
3	279.484810
4	279.597091
5	279.743216

Table 14
M obtained by GA-clustering algorithm for five different initial populations for *Crude Oil* after 100 iterations when $K = 3$

Initial population	GA-clustering
1	278.965150
2	278.965150
3	278.965150
4	278.965150
5	278.965150

Implementation results

Table 9
$\mathcal{M}$ obtained by K-means algorithm for five different initial configurations for *Vowel* when $K = 6$

Initial configuration	K-means
1	157460.164831
2	149394.803983
3	161094.118096
4	149373.097180
5	151605.600107

Table 10
$\mathcal{M}$ obtained by GA-clustering algorithm for five different initial populations for *Vowel* after 100 iterations when $K = 6$

Initial population	GA-clustering
1	149346.490128
2	149406.851288
3	149346.152189
4	149355.823103
5	149362.780998

Table 15
$\mathcal{M}$ obtained by GA-clustering algorithm for five different initial populations for *Vowel* after 500 iterations when $K = 6$

Initial population	GA-clustering
1	149344.229245
2	149370.762900
3	149342.990377
4	149352.289363
5	149362.661869

Conclusion

- The results show that the GA-clustering algorithm provides a performance that is significantly superior to that of the K-means algorithm for these data sets