STUDIA UNIV. BABEŞ-BOLYAI, INFORMATICA, VOLUME XLIII, NUMBER 2, 1998

CLUSTER PROTOTYPES SELECTION BY GENETIC CHROMODYNAMICS

D. DUMITRESCU AND ÉVA KOVÁCS

Abstract. Genetic Chromodynamics approach (Dumitrescu and Bodrogi, 1998) is considered for prototype selection in a data set. The method is particularly useful when data are not from an Euclidean space. It represents a unified approach of pattern recognition. The method is independent with respect to the considered dissimilarity measure.

1.Introduction

Evolutionary Computation (see [1], [3], [5]) is a very powerful tool in clustering and pattern recognition [8]. Prototype selection using genetic algorithms has been considered by Kuncheva and Bezdek ([9], [10]). In the approach proposed in [9] the number of clusters in data set is supposed to be known.

Genetic Chromodinamics (GC) method, proposed by Dumitrescu and Bodrogi ([6]) (see also [7], [3], [5]) can be used to solve a difficult clustering problem, namely the detection of the optimal number of clusters in a data set. The main idea of the *GC* strategy is to force the formation and maintenance of sub-populations of solutions. Clustering is an important technique used in the simplification of data or in discovering some inherent structure in the set of objects. More specifically, the purpose of cluster analysis is to partition a given set of objects into a number of groups such that objects in a particular cluster are more similar to each other than objects in different clusters.

Let $X = \{x_1, x_2,..., x_p\}$ be a set of p objects, which will be clustered. If the data set X is not from the Euclidean space or the distance defined on X is not generated by a scalar product the clustering problem becomes very difficult.

Another difficult problem is to establish the optimal number of clusters present in the data set. An approach to solve the fuzzy clustering problem in a pseudometric space has been proposed in [4].

To detect the optimal number of clusters a divisive hierarchical clustering has been considered in [2].

The aim of this paper is to propose a unified approach of pattern recognition. An evolutive procedure that unifies the geometric (or decision-making) and structural (syntactic or linguistic) approaches is considered. Moreover, the procedure, which is based on prototype selection also, gives the optimal cluster number.

The initial population is $C = \{c^1, c^2, ..., c^m\}$, where $m = p + a \cdot p, a > 0$. A chromosome c^i represents a prototype. We have $c^i_k = 1$ if c^i corresponds to the chromosome x_k. For each $i, c^i = (c^i_1, c^i_2, ..., c^i_p)$ we have

a) $c^i_k = 1$, if x_k is a prototype,

$c^i_k = 0$, otherwise;

b) $\sum_{k=1}^{p} c^i_k = 1$.

Therefore a chromosome c correspond to a prototype $x_k \in X$ if the gene c_k has the value one and the other genes are zero.

2. GC for selection of cluster prototypes from X

We are searching the cluster prototypes from the *X* data set. *GC*-based optimisation techniques start with a large arbitrary population of solutions. Dimension of the solution population will decrease at each generation. There is a high probability that each new generation will contain some individuals better than the individuals in the previous generation.

Sub-populations co-evolve and will converge towards different optimal solutions. At convergence, the number of optimal solutions equals the number of sub-populations. Each final sub-population will contain only one individual, which represents the cluster prototype.

In GC approach only local chromosome interaction are allowed. The role of local interaction is:

a) to ensure early sub-population formation and stabilization;
b) to avoid massive migration between sub-population;
c) to prevent destruction of some useful sub-population;
d) to ensure a high probability of obtaining each solution.

GC stops if after a previously fixed number of generations the number and the position of the chromosomes do not change.

The fitness function is defined as:

$$f(c) = \sum_{k=1}^{p} \frac{1}{d^n(c, x_k) + A};$$

where n is a constant greater or equal to one and A is a positive parameter. We may choose

$$n=2 \text{ and}$$

$$A = \frac{k}{p} d_{av},$$

where $k>0$ and d_{av} is the average distance between the data points. If $c_d = 1$ then $d(c,x_k)$ means $d(x_d, x_k)$.

Briefly, our GC algorithm goes through the following steps:

1. Initialization.

A set of m randomly generated chromosomes is the initial population set $C = \{c^1, c^2, ..., c^m\}$, where $m = p + a \cdot p, a > 0$ and for each $i, c^i = (c^i_1, c^i_2, ..., c^i_p)$

$c^i_k = 1$, if x_k is a prototype,

$c^i_k = 0$, otherwise;

We will consider that $\sum_{k=1}^{p} c^i_k = 1$.

GA parameters are initialized .

The interaction range r is half of the average distance between the data points, $r = \frac{d_{av}}{2}$.

(2) *Crossover.*

Each chromosome c in the population $P(t)$ will be considered for crossover. According to the proposed local interaction scheme the crossover mate of the chromosome c will be chosen from the neighbourhood $V(c,r)$ of c, according to the values of the fitness function f.

Let l be a chromosome in the interaction domain $V(c,r)$ of c. The probability that l is selected as the mate of c is denoted by $p(l)$ and may be defined as:

$$p(l) = \frac{f(l)}{\sum_{a \in V(c,r)} f(a)}$$

For selecting the mate of a given chromosome we use proportional selection.

Let *a* be the selected partner of *c*. The ordered pair *(c,a)* generates a unique offspring, *d*. The first parent is dominant, whereas the second is recessive. The unique gene in offspring having the value one is obtained from the non-zero genes of the parents, taken with equal probability. In this way the condition

$$\sum_{k=1}^{p} d_k = 1$$

holds for each offspring.

(3) Survival

Each chromosome, which participates to crossover will produce, and possibly will be replaced by an offspring. Whichever is better between a dominant parent and its offspring will be included in the new generation. Every chromosome, which does not participate at the crossover, will be included in the new generation.

(4) Steps (2) to (3) are repeated until no modification will occur for M consecutive generation.

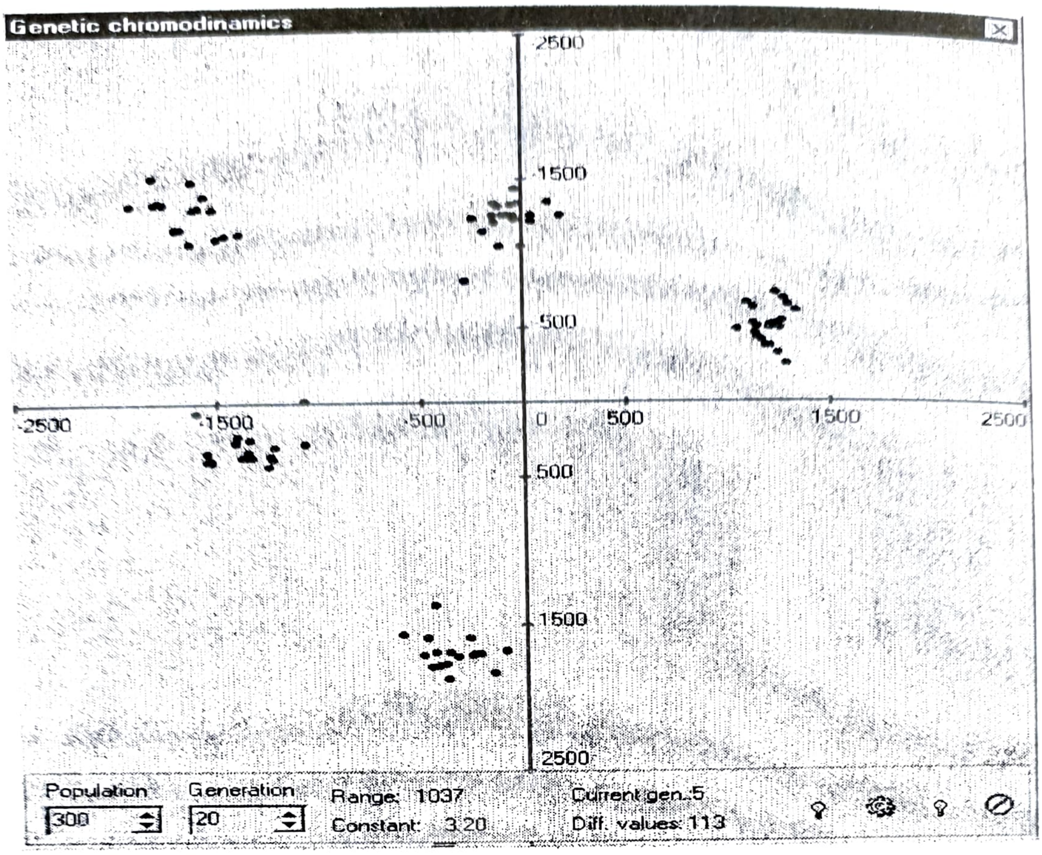
With this algorithm, we obtain the number of the optimum points, and the prototypes. Each distinct chromosome in the final population represents a prototype. With this algorithm, the population size will not decrease. But the number of distinct chromosomes decreases with the time. The values of the chromosomes will converge towards different optimum point and finally only these values will remain. As the result of the algorithm, we will obtain the prototype points. But generally the selected prototypes are not the most representative points in the respective classes.

An alternating fuzzy clustering procedure ([2], [6]) may be used for the fine tuning of the prototypes

3. Experimental results

Consider the data point depicted in Figure 1. Data set contains 300 points grouped in five clusters. The interaction range is 1037 and the constant A is 3,20.The initial population contains 591 chromosomes.

The population P(5) contains 113 different chromosomes, as indicated in Figure 2. After 10 generation 26 different chromosomes are obtained (Figure 3). After 18 generations, five distinct chromosomes are obtained (Figure 4). These chromosomes represent the final population. In Figure 5 we can see the initial population with the solutions, forming the classes.



Genetic chromodinamics

Population 300

Generation 20

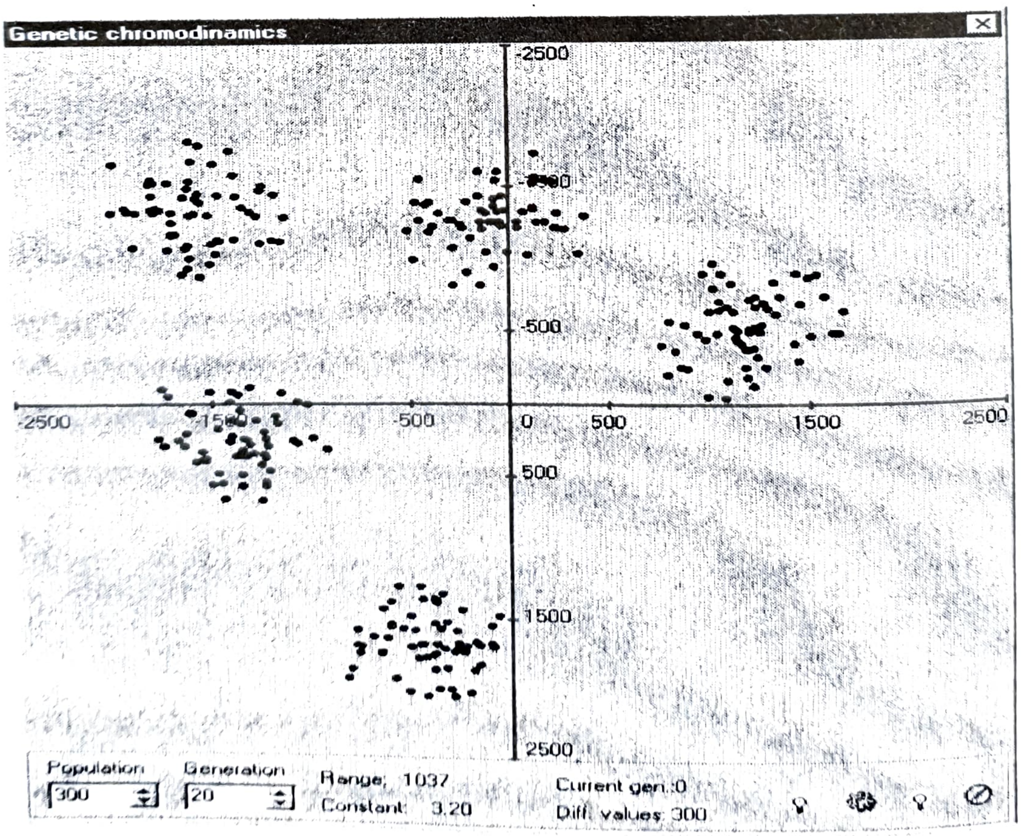
Range: 1037

Constant: 3.20

Current gen.:5

Diff. values: 113

Figure 1.



Genetic chromodinamics

Population 300

Generation 20

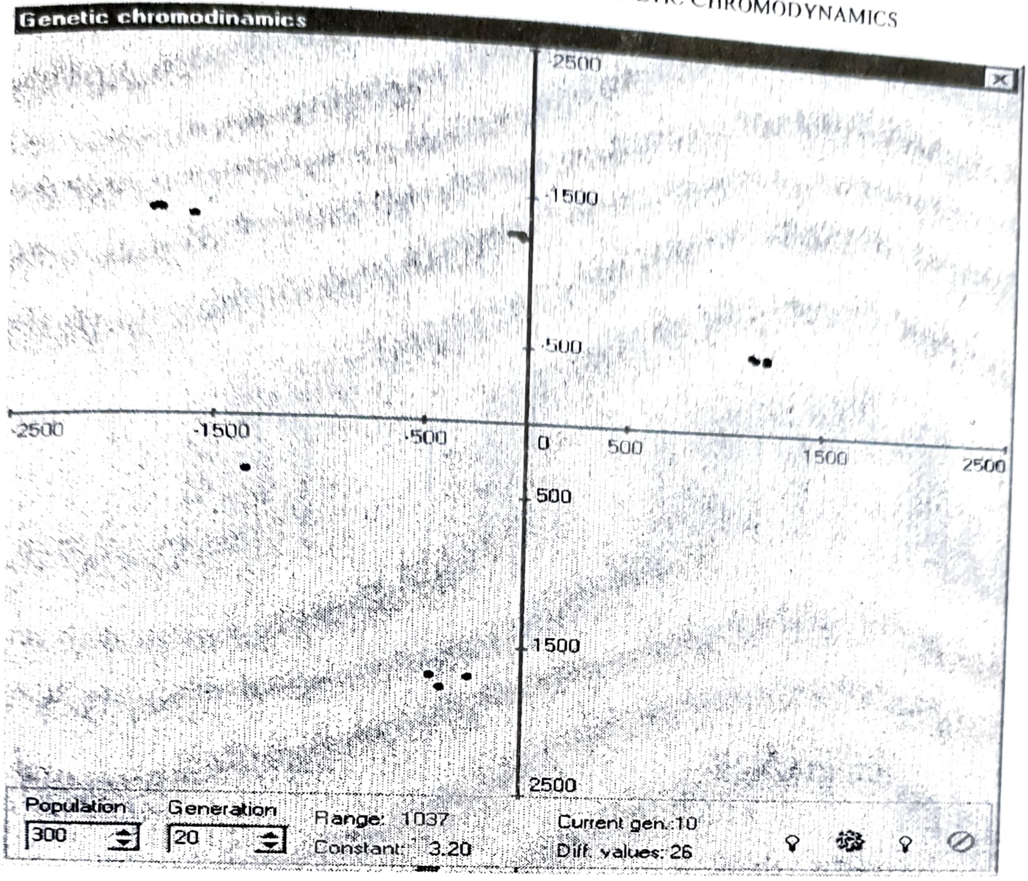
Range: 1037

Constant: 3.20

Current gen.:0

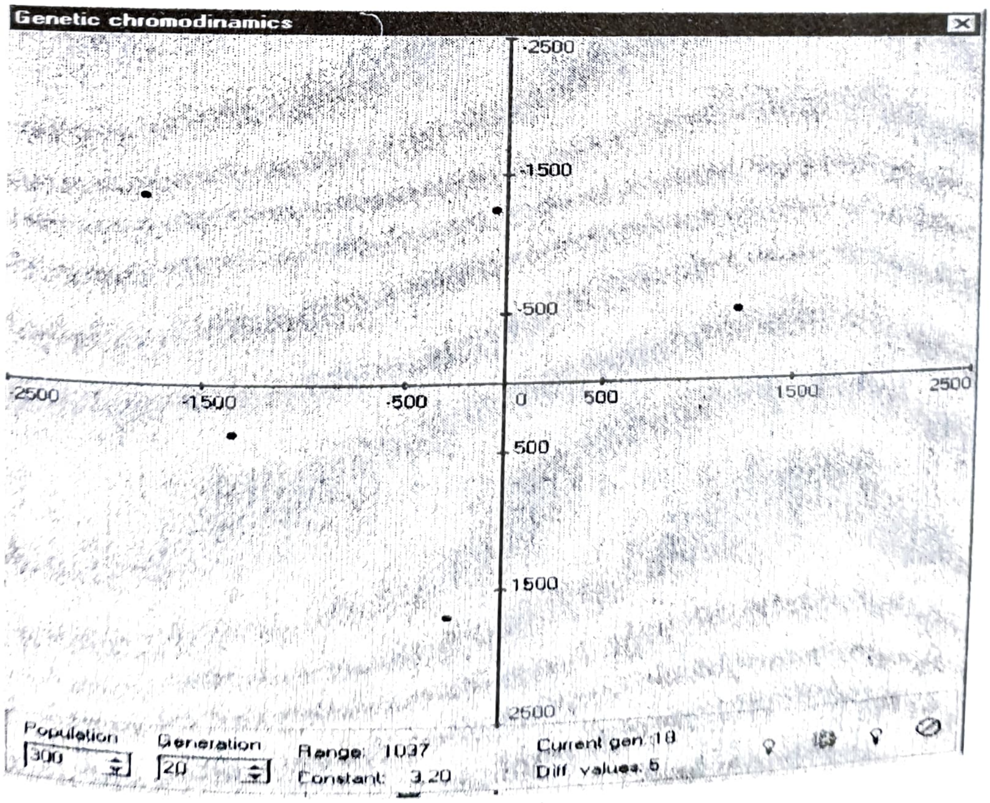
Diff. values: 300

Figure 2.



Population 300
Generation 20
Range: 1037
Constant: 3.20
Current gen.:10
Diff. values: 26

Figure 3.



Population 300
Generation 20
Range: 1037
Constant: 3.20
Current gen: 18
Diff. values: 5

Figure 4.

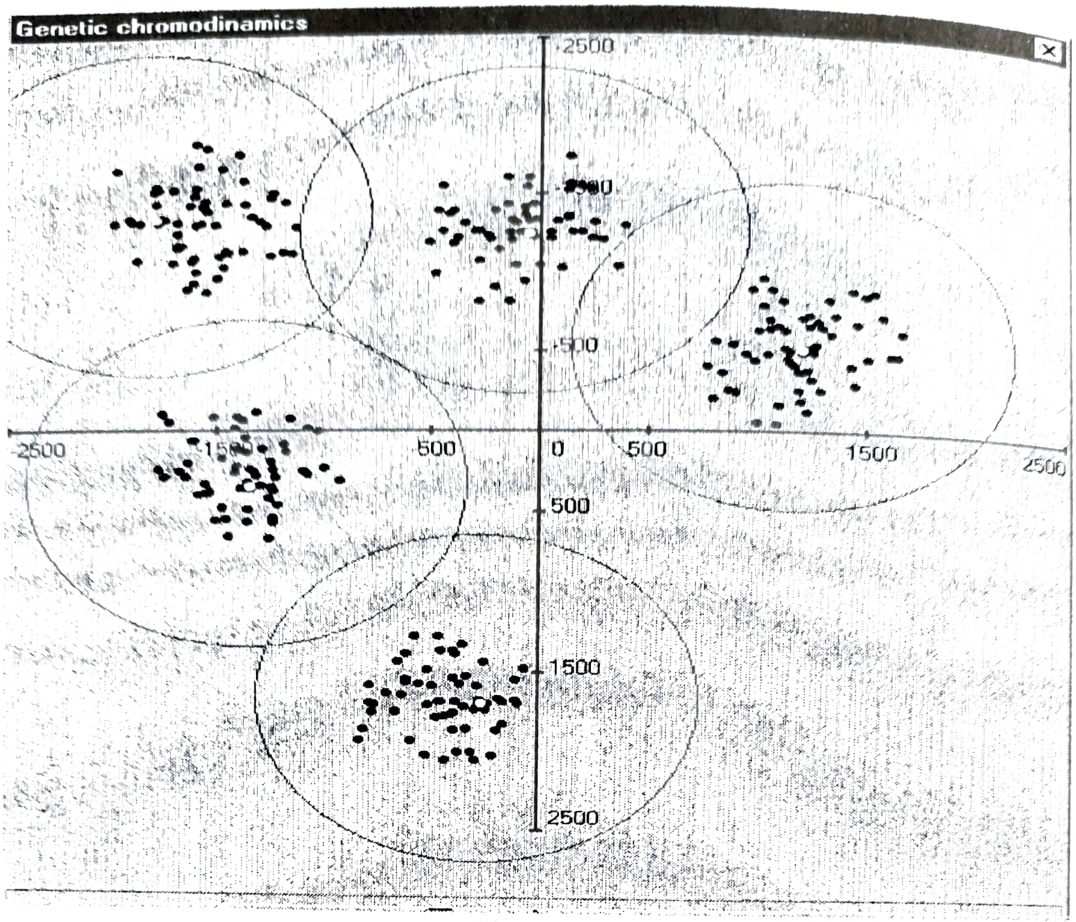


Figure 5.

REFERENCES

2. Bäck, T., Fogel, D.B., Michalewicz, Z., *Handbook of Evolutionary Computation*, Institutes of Physics Publishing, Bristol, Philadelphia and Oxford University Press, New York, Oxford, 1997.
3. Dumitrescu, D., Hierarchical pattern classification. Fuzzy Sets and Systems, 28(1988), 145-162.
4. Dumitrescu, D., Calcul evolutiv, Editura Albastra, Cluj-Napoca,1999.
5. Dumitrescu, D., Dumitrescu, A., A unified approach to fuzzy pattern recognition, European Journal of Operational Research, 96(1997), 471-478.
6. Dumitrescu, D.,Lazzerini, B.,Jain,L.C., Dumitrescu, A., Evolutionary Computation, CRC, 1999.
7. Dumitrescu, D., Bodrogi, L., A new evolutionary method and its application in clustering, Babeş-Bolyai University, Dept. of Computer Science Research Seminar, 2(1997), 115-126.
8. Dumitrescu, D., Lazzerini, B., Marcelloni, F., Ristoiu, D., Dumitrescu, A., Genetic Chromodynamics, (to be submitted).
9. Bezdek, J.C., Hathaway, R.J., Optimization of Fuzzy Clustering Criteria Using Genetic Algorithms ,First IEEE Conf. Evolutionary Computing, Orlando, FL, 1994, 589-594.
10. Kuncheva, L.I., Bezdek, J.C., Selection of Cluster Prototypes from Data by a Genetic Algorithm ,EUFIT '97, Aachen 1997, 1683-1688.
11. Kuncheva, L.I, Bezdek, J.C., Nearest Prototype Classification: Clustering, Genetic Algorithms, or Random Search? IEEE Trans SMC, Part C, 28 (1998), 160-164.

Babeş-Bolyai University, Faculty of Mathematics and Informatics, Department of Computer Science, RO 3400 Cluj-Napoca, str. M. Kogalniceanu 1, România.

E-mail address: ddumitr@cs.ubbcluj.ro

Babeş-Bolyai University, Faculty of Mathematics and Informatics, RO 3400 Cluj-Napoca, str. M. Kogalniceanu 1, România.

E-mail address: ke382@scs.ubbcluj.ro