Data Mining for Genetics: A Genetic Algorithm Approach.

Article · January 2008

Source: DBLP

CITATIONS
4

READS
1,369

2 authors:

Dr. Madhu G
VNR Vignana Jyothi Institute of Engineering & Technology
14 PUBLICATIONS **164** CITATIONS
SEE PROFILE

Keshava Reddy
Jawaharlal Nehru Technological University, Anantapur
101 PUBLICATIONS **177** CITATIONS
SEE PROFILE

Data Mining for Genetics: A Genetic Algorithm Approach

G. Madhu, Dr. Keshava Reddy E.
Dept of Mathematics,J.B. Institute of Engg. & Technology,
Yenkapally, R.R.Dist Hyderabad-500075, INDIA, A.P
Dept of Mathematics,Jawaharlal Nehru Technological University,
Ananthapur-515002,INDIA, A.P
madhumatica@hotmail.com, keshava_e@redifmail.com

Abstract

MINING biological data is an emerging area of intersection between data mining and bioinformatics. Bio-informaticians have been working on the research and development of computational methodologies and tools for expanding the use of biological, medical, behavioral, or health-related data. Biological data mining aims to extract significant information from DNA, RNA and proteins. Many biological processes are not well-understood Biological knowledge is highly complex. In this paper we discussed in discovering genetic features and environmental factors that are involved in multifactorial diseases such as (obesity, diabetes). To exploit this data, data mining tools are required and we using a specific genetic algorithm.

1. Introduction

Since the emergence of modern computers in the 1940s and 1950s, the ideas of artificial intelligence have captured the imagination of many computer scientists. Many fields of study have arisen in the pursuit of these ideas. One of the these fields, as found in the areas of computer science and engineering, is termed "evolutionary computation," the use of self-evolving strategies in problem solving, and one tool used in evolutionary computation is the *genetic algorithm* (GA). The basic concepts behind the theories of genetic algorithms were examined by John H. Holland in the mid 1970s. Genetic algorithm is an optimization technique that is a general and flexible approach to searching for optimal solutions in large, complex space [David E. Goldberg 1989]. A typical genetic algorithms approach uses an array of Booleans to represent an organism (each Boolean is considered a gene). For example of an organism is:

| 1 | 0 | 1 | 1 | 1 | 0 | 1 | ..... | 0 | 0 | 1 |

1 2 3 4 5 6 7 ……..n-2 n-1 n

This heuristic approach has been chosen as the number of features to consider is large (up to 3654 for biological data we are studying). Data provided indicates for pairs of affected individuals of a same family their similarity at given points (locus) of their chromosomes (see in fig 1).

For the first phase, the feature selection problem, we use a genetic algorithm (GA). To deal with this very specific problem, some advanced mechanisms have been introduced in the genetic algorithm such as sharing, random immigrant, dedicated genetic operators and a particular

A group of 'genes' is called a "Chromosome". The specific position in a chromosome on which a gene is located is referred to as its "locus". Chromosome is represented in a matrix where

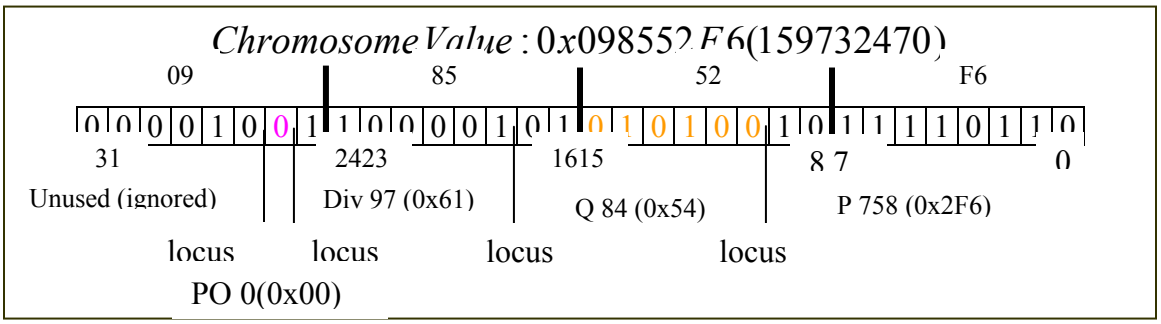

Figure 1. Chromosome example

each locus is represented by a column and each pairs of individuals considered by a row. The objective is first to isolate the most relevant associations of features, and then to class individuals that have the considered disease according to these associations. Then, the second phase, a clustering based on the features selected during the previous phase, will use the clustering algorithm *K*-means, which is very popular in clustering. Data mining can be used with Genetic algorithms together to form a method which often leads to the result, even when other methods would be too expensive or time-demanding. Now let us show this in the form of an algorithm: a) the input of the algorithm is a collection of data- training set. b) This set is encoded into a structure capable of being used for genetic algorithms. c) Encoding particular data types will be described later in detail. d) An initial population is randomly created. e) The evolution process brings out set of dependency rules or class models as its final result.

1.1 Fundamental Components Genetic Algorithms

In the ensuing sections we discuss main functional components of genetic computing such as encoding and decoding, selection, crossover, and mutation. We illustrate them with the aid of a simple rule-based example.

Encoding and Decoding

The machinery of encoding is aimed at transforming the original problem into a format amenable to genetic computations. The "inverse" transformation is realized by the decoding mechanism that allows us to move from the GA search space back to the original search space. In general in the encoding- decoding scheme we can witness three possible scenarios, as shown in Fig 2. One-one mapping, n-to- one mapping and 1-to- n mapping.

2. The Basis for the Approach

2.1 A Genetic Algorithm for a Feature selection problem

Feature selection methods attempt to find the best subset of size'd' of feature of the original N features. "Best" is typically defined as that subset of cardinality'd' which gives the best classification. In the first Phase our algorithm deals with isolating the very few relevant features from the large data set. This is not exactly the classical feature selection problem known in Data mining as in [J.Yang et al 1998]. For example around 90% of features are selected. Here, we have the idea that less than 9% of the features have to be selected problem. Our algorithm has different Phases. It proceeds for fixed number of generations. A chromosome, here, is a string of bits whose size corresponds to the number if features. A 0 or 1 at position *i*, indicates whether the feature *i* is selected (1) or not (0).

The genetic algorithm-based feature-selection mechanisms such as a correlation- based heuristic and decision-tree wrapper approach are independently used to evaluate the quality of the genetics. The analysis of the outputs, i.e., frequency, results in the identification of the significant genetics for both drug and placebo sets. With the GA as a global search tool, feature selection can be performed using two approaches, namely filter and wrapper search [Kohavi. R 1997][Hall MA, et al 1999]. It selects a feature if it correlates with the decision outcome but not to any other feature that has already been selected. Partitioning the data into drug and placebo sets along with the decision forms the initial step of the genetic algorithm genetic selection (GAGS) (Step 1, fig 3). The drug data set with n features and m observations (subjects) is evaluated using GAGS approaches step 2 in fig 3.

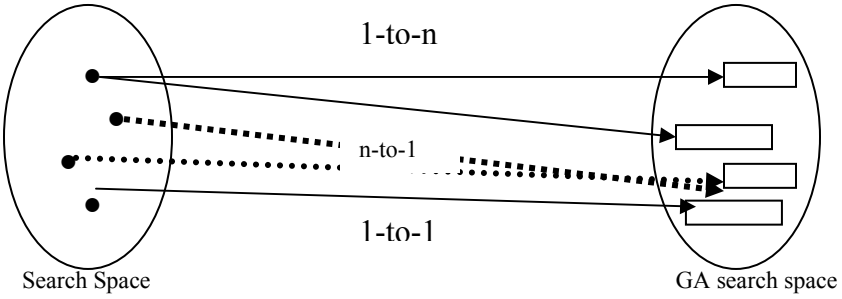

Figure 2. Encoding- decoding: mapping scenarios

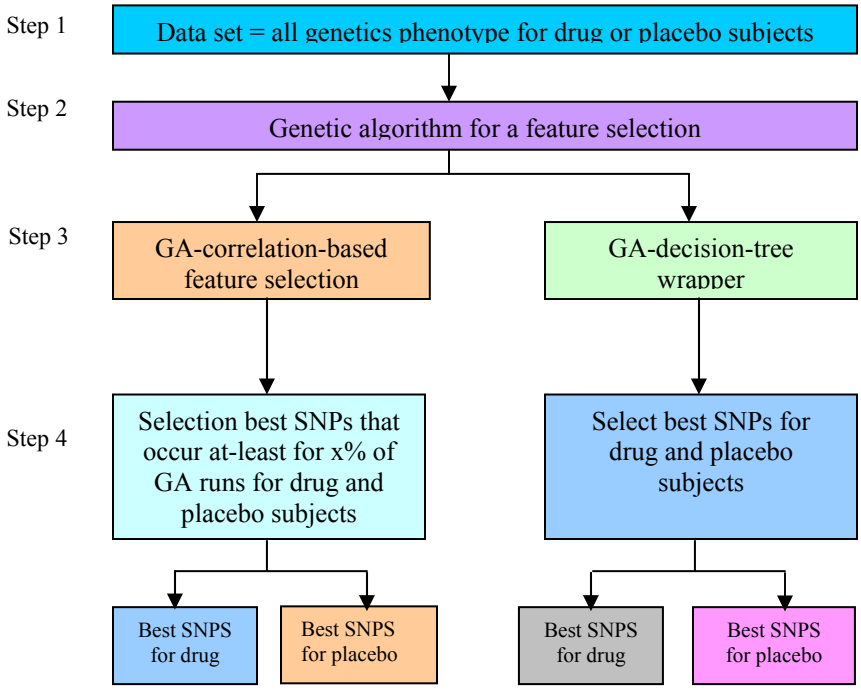

Figure 3. Genetic algorithm for a feature selection.

2.2 Fundamental Components of Genetic Algorithms

These operators allow Genetic Algorithms (GAs) to explore the search space. However, operators typically have destructive as well as constructive effects. They must be adapted to the problem.

Crossover: A one point crossover identifies two strings of the population and randomly selects a position in the strings at which they interchange their content. Fig 3 illustrates the concept of crossover in more depth. In this case the crossover happens at the fifth bit producing two new off springs-two binary strings of the form 10100100 and 01011011. The intensity of crossover is characterized in terms of the probability at which the elements of the strings are affected. The higher the probability, the more individuals are affected by the crossover.

We use a Subset Size-Oriented Common Feature Crossover Operator (SSOCF) [C. Emmanouilidis, 2000], which keeps useful informative blocks and produces offsprings which have the same distribution than the parents. Offsprings are kept, only if they fit better than the least good individual of the population. Features shared by the 2 parents are kept by offsprings and the non shared features are inherited by offsprings corresponding to the i^{th} parent with the probability $(n_i - n_c/n_u)$ where n_i is the number of selected features of the i^{th} parent, n_c is the number of commonly selected features across both mating partners and n_u is the number of non-shared selected features.

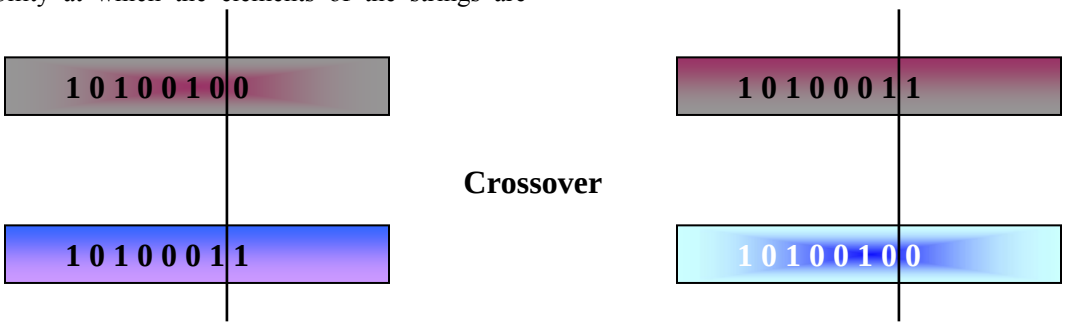

Figure 3. An example of a Single –point crossover

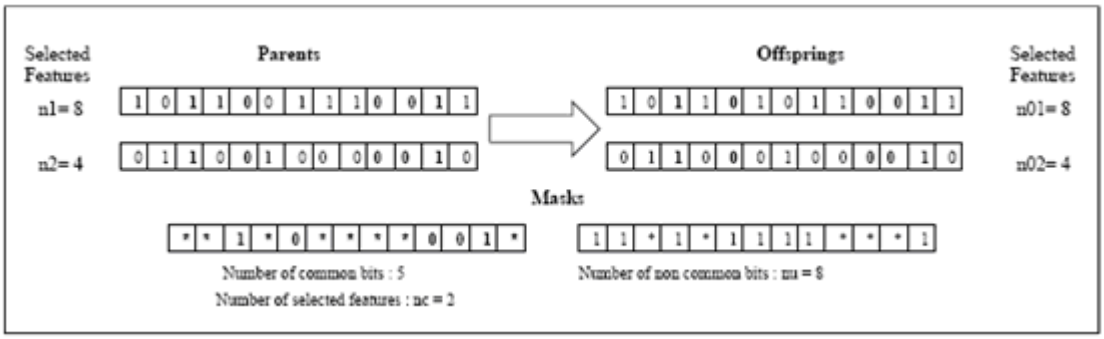

Figure 4. The SSOCF crossover Operator.

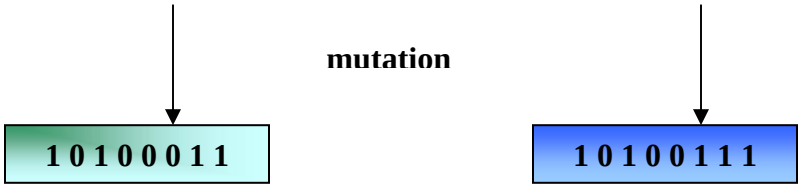

Figure 5. Mutation mechanisms in binary strings.

Mutation: The mutation operator is an example of an operator adding an extra diversity of stochastic nature. In binary strings this mechanism is implemented by flipping the values of some randomly selected bits, see fig 5. Again the mutation rate is related to the probability at which the individual bits become affected. For instance, a mutation rate of 5% when applied to a population of 500 strings each being 20 bits long means 5% of 1,000 bits being changed.

The mutation is an operator which allows diversity. During the mutation stage, a chromosome has a probability p_{mut} to mutate. If a chromosome is selected to mutate, we choose randomly a number n of bits to be flipped then n bits are chosen randomly and flipped. In order to create a large diversity, we set p_{mut} around 10% and $n \in [1, 5]$.

Selection

We implement a probabilistic binary tournament selection. Tournament selection holds n tournaments to choose n individuals. Each tournament consists of sampling 2 elements of the population and choosing the best one with a probability $p \in [0.5, 1]$.

3. Adaptations and Mechanisms

3.1 The chromosomal distance:

In this module the biologist experts indicate that a gene is correlated with its neighbors situated on the same chromosome at a distance smaller than σ equals to 20 CMorgan (a measure unit). So in order to compare two individuals, we create a specific distance which is a kind of bit to bit distance where not a single bit i is considered but the whole window $(i-\sigma, i+\sigma)$ of the two individuals are compared. If one and only one individual has a selected feature in this window, the distance is increased by one.

3.2 Fitness function:

The fitness function becomes crucial in an evaluation of the performance of the individual chromosome. In this, problem the fitness function we developed refers to the support notion, for an association, which, in data mining, denotes the number of times an association is met over the number of times at least one of the members of the association is met. The function is composed of two parts. The first one favours for a small support a small number of selected features because biologists have in mind that associations will be composed of few features and if an association has a bad support, it is better to consider less features (to have opportunity to increase the support). The second part, the most important (multiplied by 2), favours for a large support a large number of features because if an association has a good support, it is generally composed of few features and then we must try to add other features in order to have a more complete association. What is expected is to favour good associations (in term of support) with as much as features as possible. This expression may be simplified, but we let it in this form in order to identify the two terms.

$$F = \left[\left((1 - S) \times \frac{\frac{T}{10} - 10 \times SF}{T} \right) + 2 \times \left(S \times \frac{\frac{T}{10} - 10 \times SF}{T} \right) \right]$$

Where support $S = \left| \frac{A \cap B \cap C \ldots\ldots}{A \cup B \cup C \ldots\ldots} \right| where A, B, C \ldots\ldots are the selected features.$

T = Total Number of features, and SF = Number of selected significant features.

4. The clustering phase (using K-means Algorithm)

Let us briefly examine a simple but useful algorithm, namely, the k-means algorithm and its generated form, k-mode algorithm. Both can be stated as an optimization problem, which can be considered as an alternative way of dealing with *uncertainty*. The *k*-means algorithm is an iterative procedure for clustering which requires an initial classification of the data. The *k*-means algorithm proceeds as follows: it computes the center of each cluster, and then computes new partitions by assigning every object to the cluster whose center is the closest (in term of the Hamming distance) to that object. This cycle is repeated during a given number of iterations or until the assignment has not changed during one iteration [N. Monmarch´e, 1999]. Since the number of features is now very small, we implement a classical *k*-means algorithm [Z.Chen 2001] widely used in clustering, and to initialize the procedure we randomly select initial centers.

Algorithm K-means Clustering

Step 1: Choose k cluster centers to coincide with k randomly-chosen patterns of k

Randomly defined points inside the hyper volumn containing the pattern set.

Step 2: Assign each pattern to the closest cluster center.

Step 3: Re-compute the cluster centers using the current cluster memberships.

Step 4: If a convergence criterion is not met, go to step 2. Typical convergence criteria are: no (or minimal) reassignment of patterns to new cluster centers, or minimal decrease in squared error.

5. Experimental Studies

Experiments have been first executed on an artificial database in order to validate the method. This database was constructed for the workshop challenge GAW11 (Genetic Analysis Workshop) which was hold in 1998 and was organized by Dr David A. Greenberg1 to test different data mining methods. This database is a public one. This is an artificial database constructed to be close to real problems and we know, by construction, the relevant associations of features which can influence the disease. Results to obtain are associations A+B+D and $C+E_1$. This test base is composed of 500 features and 169 pairs of individuals. For ten runs, we wanted to know how many times associations were discovered by the GA. We noted the following results:

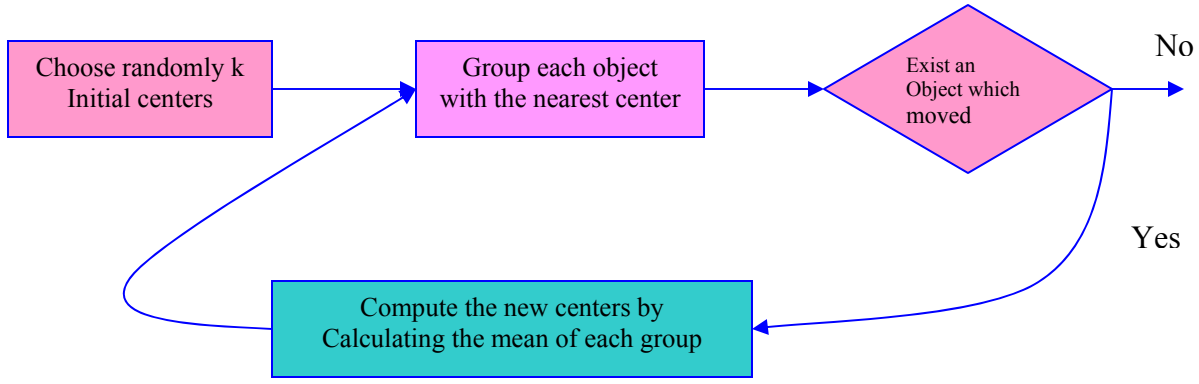

Figure 5. The *k*-means algorithm

Association	A+B	A+D	B+D	C+E_1
	100%	50%	20%	10%

The first phase was able to discover real interactions of locus. Some of them are more difficult than other to find. Then, we ran the *k*-means algorithm with the results of the GA. We gave 13 features selected by the GA, instead of the initial 500 to the *k*-means algorithm. The *k*-means algorithm helps us to discover associations genes-genes and genes-environmental factors.
We have experimented the classical *k*-means algorithm without any feature selection. The execution time was very large (over 7500 minutes) and results can not be interpreted (we didn't know which were the features involved in the disease) so the feature selection phase is required. With the feature selection, the time of execution of *k*-means had decreased to 1 minute and the results are exploitable. We present here clusters obtained with *k*=2 and their number of occurrences:

(A B D) (E_1 C)	(A B D) (E_1 C D)	(A B) (E_1 D C)	(A B) (E_1)	(E_1 A D B) (E_1 C B)	(A B D) (E_1 D)
4	1	1	2	1	1

This table shows that the *k*-means algorithm using results of the GA is able to construct clusters very closely related to the solution presented in results of the workshop. Moreover this solution has been exactly found 4 times over 10 of executions.
First, we tested the performances of the method in term of size of problems it can deal with. It appeared that the execution time grows linearly with the number of features and the number of pairs. So the method is able to deal with very large size problem. Then, we ran several times the algorithm. The genetic algorithm managed to select interesting features and the *k*-means algorithm was able to class pairs of individuals according to these features and to conform interesting associations of features.

6. Conclusions

In this paper we presented a genetic algorithm which for a particular feature selection problem encountered in genetic analysis of different diseases. The specificatties of this problem is that we are not looking for single feature but for several associations of features that may be involved in the studied disease. Results are promising for biologists as the algorithm seems to be robust and to be able to isolate interesting associations. Future work using Genetic Algorithm for Feature Extraction based K-nearest-neighbor algorithm to evaluate the effectiveness of the weight vector in increasing separation between known pattern classes.

7. References

[1] C. Bates Congdon. A comparison of genetic algorithm and other machine learning systems on a complex classification task from common disease research. PhD thesis, University of Michigan, 1995.

[2] Bradley, P.S.; Fayyad, U.; Reina, C (1998) Proceedings Fourth International Conference on Knowledge Discovery and Data Mining, Agrawal, R., Stolorz, P.(Eds), pp. 9-15.

[3] David E. Goldberg (1989) Genetic Algorithms in Search, Optimization, and Machine Learning.

[4] Cios et al., (1998) K. Cios, W. Pedrycz, and R. Swiniarski. Data Mining Methods for Knowledge Discovery, Kluwer, 1998.

[5] Z.Chen. (2001). Data mining and Uncertain Reasoning: An Integrated approach, Wiley.

[6] C. Emmanouilidis, A. Hunter, and J. MacIntyre. A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator. In Congress on Evolutionary Computing 2000, volume 2, pages 309–316. CEC, 2000.

[7] J. Horn, D.E. Goldberg, and K. Deb. Implicit niching in a learning classi.er system : Nature's way. Evolutionary Computation, 2(1):37–66, 1994.

[8] Hall MA, Smith LA. Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper. In: Kumar A, Russell I, editors. Proceedings of the Florida Artificial Intelligence Research Symposium, Orlando, Florida. Menlo Park, CA: AAAI Press; 1999. p. 235—239. ISBN: 1577350804.

[9] Kohavi R, John GH. Wrappers for feature subset selection. Artif Intell 1997;97(1—2):273—324

[10] S.W. Mahfoud. Niching Methods for Genetic Algorithms. PhD thesis, University of Illinois, 1995.

[11] N. Monmarch´e, M. Slimane, and G. Venturini. Antclass : discovery of cluster in numeric data by an hybridization of an ant colony with the kmeans algorithm. Technical Report 213, Ecole d'Ing´enieursen Informatique pour l'Industrie (E3i), Universit´e de Tours, Jan. 1999.

[12] M. Pei, E.D. Goodman, and W.F. Punch. Feature extraction using genetic algorithms. Technical report, Michigan State University : GARAGe, June 1997.

[13] M. Pei, E.D. Goodman, W.F. Punch, and Y. Ding. Genetic algorithms for classi.cation and feature extraction. In

Annual Meeting : Classi.cation Society of North America, June 1995.

[14] M. Pei, M. Goodman, and W.F. Punch. Pattern discovery from data using genetic algorithm. In Proc of the .rst Paci.c-Asia Conference on Knowledge Discovery and Data Mining, Feb. 1997.

[15] J. Yang and V. Honoavar. Feature Extraction, Construction and Selection : A data Mining Perspective, chapter 1: Feature Subset Selection Using a Genetic Algorithm, pages 117–136. H. Liu and H. Motoda Eds, massachussetts : kluwer academic publishers edition, 1998.