Journal of Computer Science 3 (9): 723-725, 2007
ISSN 1549-3636

Feature Selection in Data-Mining for Genetics Using Genetic Algorithm

V.N. Rajavarman and S.P. Rajagopalan
School of Computer Science and Engineering, Dr. M.G.R. University,
Chennai, Tamilnadu, India

Abstract: We discovered genetic features and environmental factors which were involved in multifactorial diseases. To exploit the massive data obtained from the experiments conducted at the General Hospital, Chennai, data mining tools were required and we proposed a 2-Phase approach using a specific genetic algorithm. This heuristic approach had been chosen as the number of features to consider was large (upto 3654 for biological data under our study). Collected data indicated for pairs of affected individuals of a same family their similarity at given points (locus) of their chromosomes. This was represented in a matrix where each locus was represented by a column and each pairs of individuals considered by a row. The objective was first to isolate the most relevant associations of features and then to class individuals that had the considered disease according to these associations. For the first phase, the feature selection problem, we used a genetic algorithm (GA). To deal with this very specific problem, some advanced mechanisms had been introduced in the genetic algorithm such as sharing, random immigrant, dedicated genetic operators and a particular distance operator had been defined. Then, the second phase, a clustering based on the features selected during the previous phase, will use the clustering algorithm k-means.

Key words: Crossover, mutation, selection, fitness function, random immigrant, k-means algorithm

INTRODUCTION

The first phase of our algorithm deals with isolating the very few relevant features from the large set. This is not exactly the classical feature selection problem known in Data mining as, in[9], for example around 50% of features are selected. Here, we have the idea that less than 5% of the features have to be selected. But this problem is close from the classical feature selection problem and we will use a genetic algorithm as we saw they are well adapted for problems with a large number of features[6,7,8]. We present here the main characteristics and adaptations we made to deal with this particular feature selection problem. Our genetic algorithm has different phases. It proceeds for a fixed number of generations. A chromosome, here, is a string of bits whose size corresponds to the number of features. A 0 or 1, at position i, indicates whether the feature i, is selected (1) or not (0).

The genetic operators: These operators allow GAs to explore the search space. However, operators typically have destructive as well as constructive effects. They must be adapted to the problem.

Crossover: We use a Subset Size-Oriented Common Feature Crossover Operator (SSOCF)[2] which keeps useful informative blocks and produces offspring's 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.

Mutation: 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 € [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 €¸[1,5].

Corresponding Author: V.N. Rajavarman, Dr. M.G.R. University, Maduravoyal, Chennai, India, 600095

METERIALS AND METHODS

Specific adaptations and mechanisms: The chromosomal distance (A distance adapted to the problem): The biologist experts indicate that a gene is correlated with its neighbours 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-σ, i+σ) 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.

The fitness function: 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 favours 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.

$$F = ((1-S) * (T/10-10*SF) /T)+2*(S* (T/10-10*SF)/T$$

Where:

$S = \left|(\alpha\cap\beta\cap\gamma\ldots)/(\alpha\cup\beta\cup\gamma\ldots)\right|$ where $\alpha,\beta,\gamma\ldots$ are the selected features

T = Total number of features,
SF = Number of selected significant features

Sharing: To avoid premature convergence and to discover different good solutions (different relevant associations of features), we use a niching mechanism. A comparison of such mechanisms has been done in[4]. Both crowding and sharing give good results and we choose to implement the fitness sharing[3].

The objective is to boost the selection chance of individuals that lie in less crowded area of the search space. We use a niche count that measures of how crowded the neighborhood of a solution is. The distance δ is the chromosomal distance adapted to our problem presented before. The fitness of individuals situating in high concentrated search space regions is degraded and a new fitness value is calculated and used, in place of the initial value of the fitness, for the selection.

The sharing fitness $f_{sh}(i)$ of an individual i, where n is the size of the population,$\alpha_{sh} = 1$ and $\sigma_{sh} = 3$), is:

$$\mathrm{fsh}(i) = \frac{F(i)}{\sum_{j=1}^{n} \mathrm{Sh}(\delta(Ii, Ij))}$$

where:

$$\mathrm{Sh}(\delta(Ii, Ij)) = \left\{ \begin{matrix} 1 - \left(\frac{\delta(Ii, Ij)}{\sigma sh} \right)^{a,h} & \text{if}\delta(Ii,Ij) \ \langle \sigma sh \\ o & \text{else} \end{matrix} \right\}$$

Random immigrant: Random Immigrant is a method that helps to maintain diversity in the population. It should also help to avoid premature convergence[1]. We use random immigrant as follows: if the best individual is the same during N generations, each individual of the population, whose fitness is under the mean, is replaced by a new randomly generated individual. When random immigrant is done, we add an extra step in our algorithm.

The clustering phase

Use of k-means algorithm: 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, 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[5].

Since the number of features is now very small, we implement a classical k-means algorithm widely used in clustering and to initialise the procedure we randomly select initial centers.

Experiments: Validation on an artificial database: In order to validate the method, experiments have been first executed on an artificial database constructed to be close to real problems, which is a public one. we know, by construction, the relevant associations of features which can influence the disease.

Table 1: Association

Association	α+β	α+δ	β+δ	ϒ+ω
	100%	50%	20%	10%

Table 2: Occurrence

(α β δ)	(α β δ)	(α β)	(α β)	(ω α δ β)	(α+β+δ)
(ω ϒ)	(ω ϒ δ)	(ω δ ϒ)	(ω)	(ω ϒ β)	(ω δ)
4	1	1	2	1	1

Results to obtain are associations $\alpha+\beta+\delta$ and $\Upsilon+\omega$. This test base is composed of 491 features and 165 pairs of individuals. For ten runs, we wanted to know how many times associations were discovered by the GA. We noted the following results in Association Table 1.

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 11 features selected by the GA, instead of the initial 491 to the k-means algorithm.

RESULTS AND DISCUSSION

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 (Occurrence Table 2). This Table 2 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.

Experiments are executed on real data provided by the General Hospital at Chennai for the study of diabetes. The dataset is composed of 1179 pairs of individuals who have diabetes. The biologists take 3552 points of comparison and 2 covariables (age at onset, the age of the individual when diabetes was diagnosed and BMI Body Mass Index, which is a measure of obesity). The data are confidential and we can not give any biological results here, but here is some aspects:

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 confirm interesting associations of features.

CONCLUSION

We present here a genetic algorithm dedicated for a particular feature selection problem encountered in genetic analysis of different diseases. The specificities 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. Those associations have now to be studied and validated by the biologists.

REFERENCES

1. Bates Congdon, C., 2002. A comparison of genetic algorithm and other machine learning systems on a complex classi. Cation task from common disease research. Ph.D Thesis, University of Michigan.
2. Emmanouilidis, C., A. Hunter and J. MacIntyre, 2000. A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator. In Congress on Evolutionary Computing, 2: 309-316.
3. Horn, J., D.E. Goldberg and K. Deb, 1994. Implicit niching in a learning classifier system: Nature's way. Evolutionary Computation, 2: 37-66.
4. Mahfoud,S.W., 2004. Niching Methods for Genetic Algorithms. Ph.D Thesis, University of Illinois.
5. Monmarch´e, N., M. Slimane and G. Venturini, 2001. Antclass: discovery of cluster in numeric data by an hybridization of an ant colony with the kmeans algorithm. Technical Report 213, Ecole d'Ing´enieurs en Informatique pour l'Industrie (E3i), Universit´e de Tours.
6. Pei, M., E.D. Goodman and W.F. Punch, 1997. Feature extraction using genetic algorithms. Technical report, Michigan State University: GARAGe.
7. Pei, M., E.D. Goodman, W.F. Punch and Y. Ding, 1995. Genetic algorithms for classi.cation and feature extraction. In Annual Meeting: Classi.cation Society of North America.
8. Pei, M., M. Goodman and W.F. Punch, 1997. Pattern discovery from data using genetic algorithm. In Proc. rst Paci. c-Asia Conference on Knowledge Discovery and Data Mining.
9. Yang, J. and V. Honoavar, 2005. Feature Extraction Construction and Selection: A data Mining Perspective, chapter 1: Feature Subset Selection Using a Genetic Algorithm, H. Liu and H. Motoda Eds, massachussetts: kluwer academic publishers Ed., pp: 117-136.