ISSN:0975-9646

GA-Correlation Based Rule Generation for Expert Systems

Harsh Bhasin[#1], Supreet Singh[*2]
[#]*Computergrad.comy*
Faridabad, India
[*]*Student, CITM*
Faridabad, India

***Abstract*—Rule generation in an expert system requires heuristics and selection procedures which are not just accurate but are also efficient. This premise makes Genetic Algorithms (GAs) a natural contender for the rule selection process. The work analysis the previous attempts of applying GAs to rule selection and proposes major changes in the clustering algorithms for rule generation. The representations of a cluster, the formula for probability and the term distance have been modified. Also the coefficient of auto correlation and not mean is taken as the deciding factor for an item to be in the cluster. The work has been implemented and analyzed and the results obtained are encouraging.**

***Keywords*— Genetic Algorithms, Expert System, Artificial Intelligence, Clustering.**

I. INTRODUCTION

Expert system (ES) is a computer program that simulates thought process of a human expert to solve complex decision problem [1], with a strong belief that the growth of ES is bound to happen and newer once will have immense data thus requiring a novel search process. The work presents the use of Genetic Algorithms (GAs) for the rule selection process in an ES.

One of the most important characteristic of an ES is an interactive decision based tool that use both facts and heuristics to solve problems based on knowledge from Experts. The crux of the work is that computational or deterministic applications are not good candidates for ES Rule Generation Process. ES deals with those problems which require heuristics. The problems in which intuitions are needed, which require judgment and logical inferences are needed to be solved by an ES. The present work concentrates on automatic rule generation by clustering the data in groups which change as the ES evolves. Major changes have been proposed and analysed. The work maps the problem with the Genetic Algorithms (GAs) by taking a population consisting of chromosomes having integers as their cells.

II. EXPERT SYSTEM

It has the following essential characteristics

A. DOMAIN SPECIFICITY

While describing an ES the developer must be very clear as regards its goals. For example an ES for psychosomatic diseases cannot deal with composition of blood.

B. SPECIAL LANGUAGE

An ES is generally made in languages like prolog and lisp. The work intends to challenge this belief by proposing an ES in .NET framework. The motivation for using lisp etc was good memory procedure and extensive data manipulation routine both of which can be implemented in the framework by using C# or F#.

C. STRUCTURE

Knowledge must be organized in an accessible format. So the system has three distinct levels.

D. KNOWLEDGE BASE

Knowledge base has capacity to solve problem and it has data that is relevant to the problem domain.

E. MEMORY

It requires an efficient use of memory modules for various tasks.

F. INFERENCE ENGINE:-

They are generic control mechanism that applies axioms to knowledge base so as to solve the particular task.

A Knowledge base can in no way be considered as a database because database deals with static data whereas in Knowledge base there are strategies and which recommend directions for user enquiry. The structure of an ES is shown in Figure1.

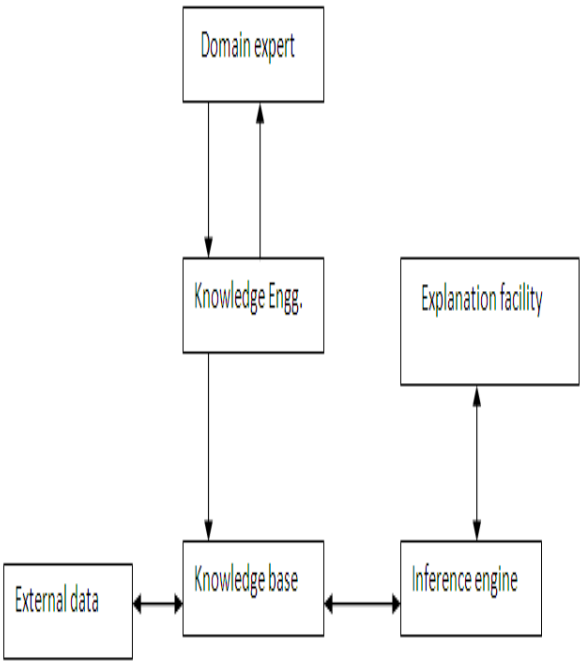

Fig. 1 Expert System

III. GENETIC ALGORITHMS

Genetic Algorithms (GAs) are search procedures to converge to optimal solution based on the theory of survival of the fittest. The basic entity of GA is chromosome. Each chromosome symbolizes a solution to the problem and is composed of a string of cells of finite length. The binary alphabet {0, 1} is often used to represent these cells but integers can be used depending on the application. The fitness value is a function or rationale against which chromosome is tested for its suitability to the problem in hand [5].

GAs transforms the population of mathematical object into new population with a better fitness using the following operators:

A. Crossover

It is a genetic operator that coalesce two chromosomes to produce a new chromosome. The child chromosome takes one section of the chromosome from each parent. The point at which chromosome is broken depends on the randomly selected crossover point [6].

The number of crossovers is determined by the crossover rate which is generally 2-5%. GAs have following type of crossovers:

Single point crossover: In this, a single crossover point is selected and is applied on two chromosome selected.

Two point crossover: In this type of crossover two crossover points are selected and the crossover operator is applied.

Uniform Crossover: In this type, bits are copied from both chromosomes uniformly.

B. Mutation

It is necessary for ensuring genetic diversity in population. GAs involves string-based modifications to the elements of a candidate solution. These include bit-reversal in bit-string GAs or shuffle and swap operators in permutation GAs [5], [6].

C. Selection

It is quantitative criterion based on fitness value to choose which chromosomes from population will go to reproduce. Intuitively the chromosome with more fitness value will be considered better and in order to implement proportionate random choice, Roulette wheel selection is used for selection [5], [6], [7].

IV. LITERATURE REVIEW

Many papers have been studied to get an idea of various clustering algorithms. The most common techniques of clustering have been discussed in this section.

A. K-means Clustering

The K-means clustering is an algorithm based on finding data clusters in a data set such that an objective function of distance measure is minimized. In the majority of the cases this variation is preferred as the Euclidean distance. The partition groups are defined by a binary membership matrix, where the element ij is 1 if the jth data point jx belongs to group i, and 0 otherwise. After the cluster centres are fixed, the minimizing Equation can be applied, which results in an element belonging to a group i if the distance is the closest centre among all centres.

In this technique the cluster centre are initialized by randomly selecting c points from among all of the data points. This is followed by determining the membership matrix U by requisite function. Then the cost function is computed. Finally, the process stops if it is below a certain tolerance value or cluster stabilizes [9].

It is a known fact that the performance of the K-means algorithm depends on the initial positions of the cluster centres, thus it is becomes essential to run the algorithm several times with a different set of initial cluster centres.

B. Fuzzy C-means Clustering

In FCM each data point belongs to a cluster to a degree of membership score. Therefore FCM utilizes fuzzy partitioning such that a given data point can belong to several groups with the degree of belongingness specified by membership grades between 0 and 1. FCM uses a cost function that is to be minimized while trying to detach the data set. The membership matrix is allowed to have elements with values between 0 and 1. The summation of degrees of belongingness of a data point to all clusters is always equal to unity [10].

C. Mountain Clustering

The mountain clustering approach is an uncomplicated technique to locate cluster centres. It is based on a density measure called the mountain function. The first step in mountain clustering involves forming a grid on the data space, where the intersections of the grid lines constitute the possible cluster centres. The second step involves making a mountain function on behalf of a data density measure [10].

D. Subtractive Clustering

The difficulty with mountain clustering is that its computation increases exponentially with the dimension of the problem. The mountain function has to be evaluated at each grid point. This problem is solved by using data points as the candidates for cluster centres. This makes the problem size the deciding factor which decides the growth and not the problem dimension. This however is a good approximation not an exact method [10].

V. PROPOSED WORK

For automatic rule generation clustering of data in groups is required. The work carries forward the notion of variable length chromosome as proposed in earlier work [2] [3].In each cluster there are many cells and number of cells varies. In our proposal a cluster may contain many cells each of which represents a number to be clustered. In our implementation the first cell indicates the number of cell in the particular cluster. This difference in the approach from the earlier one is sure to make the work efficient. The cluster is shown in Figure 2.

In the earlier works [2], [3] the distance is calculated by taking the mean as the centrality. In our implementation the initial population is generated by subtractive clustering but the betterment of the clusters is done using coefficient of autocorrelation. It is done so that ethos of population can be judge in a better way. The initialization is done by subtractive clustering, but the conviction of doing so lies in the fact that the once clusters are made then a reorder is needed to increase the cohesiveness not just to settle the set of points about the mean point.

A minor change in the formula for probability has also been proposed because the probability of finding out a data point in a cluster can be found by the following formula.

Pi = 1- disntance2/ Σ distances2

In the earlier works it was suggested that probability of a point belongs to a cluster increases as the square of distances decreases.

The above points are summarized in the algorithm given below:

Step 1: Perform subtractive clustering

Step 2: Repeat for all data points

Repeat for each cluster

Calculate the coefficient of autocorrelation of that cluster. Store each coefficient in array B [].

End loop

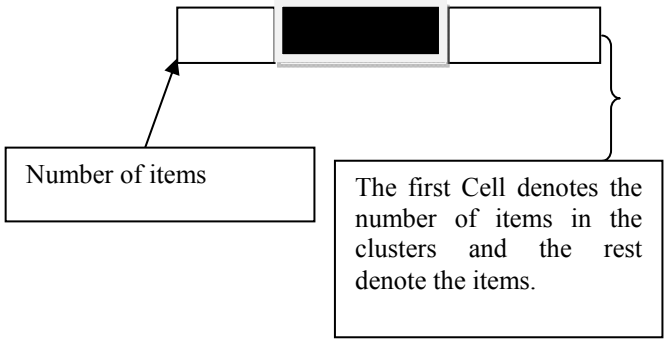

Fig. 2 Representation of Chromosomes

Step 3: Repeat

Generate a population of chromosomes as 2D array chromosomes [number of chromosomes][maximum number of cells in any cluster].

Step 4: Generate two random numbers n1 and n2, each less than the number of chromosomes.

Step 5: Select cluster number n1 and n2.

Step 6: Perform XOR-ing of chromosome n1 and chromosome n2. In the result if there is 1 and the position at which swapping is to be done is less than the value in the first bit of either of the cluster, then swap the items of cluster n1 and cluster n2 at that position.

Step 7: Recalculate the coefficient of auto correlation and update B. Until the values of B stops changing.

The process is summarized in the figure 3.

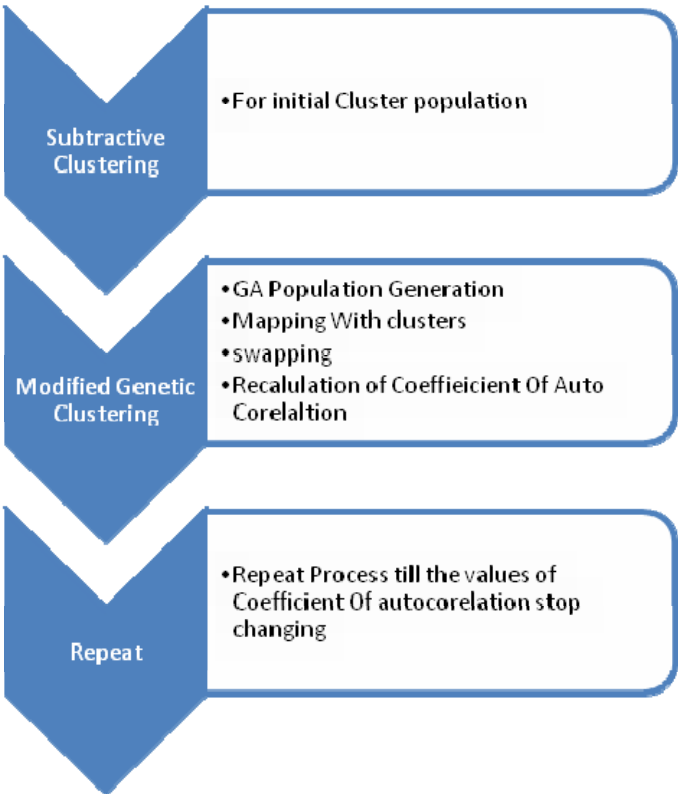

Fig. 3 Process

VI. Results and Conclusions

The work has been implemented and analysed with satisfactory results. Mean is a measure of central deviation it gives a good idea of what should be the middle point of a particular cluster but it does not tell us about the cohesiveness of the cluster. Statistically coefficient of correlation gives a better idea of the relation between the items. The work was analysed with 23 samples having 200 items each. The work proposes major changes in the present algorithm. A comparison has also been done with subtractive clustering and Genetic clustering proposed in [2], [3].

Due to limited time and recourses the work has been compared with the above said algorithms. In the future work a sample ES is to be implemented and the work proposed will be put to test in that ES.

References

[1] John C. Determan ,James A. Foster," A Genetic Algorithm for Expert System Rule Generation" ,Published in: · Proceeding ICPR '04 Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 1 - Volume 01 IEEE Computer Society Washington, DC, USA ©2004

[2] [2] Peter Ross, Dave Corne, "APPLICATION OF GENETIC ALGORITHM", 1993, Proceedings of the Sixth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems

[3] [3] Kropotov Dmitry, Vetrov Dmitry ," An Algorithm for Rule Generation in Fuzzy Expert Systems ",· Proceeding ICPR '04 Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 1 - Volume 01

[4] S.V. Wong, A.M.S. Hamouda ," Optimization of fuzzy rules design using genetic algorithm ",Published in: · Journal Advances in Engineering Software archive Volume 31 Issue 4

[5] Harsh Bhasin, Gitanjali, Harnessing Genetic Algorithm for Vertex Cover Problem, International Journal on Computer Science and Engineering (IJCSE), 4(2), 218 - 223.

[6] Randomized algorithm approach for solving *PCP Harsh Bhasin*, Nishant Gupta IJCSE 2012; 4(1):106-113. ICID: 976303.

[7] Use of Genetic Algorithms for *Finding Roots* of. Algebraic Equations. *Harsh Bhasin IJCSIT; 2(4)*

[8] Jang, J.-S. R., Sun, C.-T., Mizutani, E., "Neuro-Fuzzy and Soft Computing – A Computational Approach to learning and Machine Intelligence," Prentice Hall.

[9] Hammouda, K. (2000). A Comparative Study of Data Clustering Techniques. Design, 625, 1-21. Citeseer. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.126.3224&rep=rep1&type=pdf

[10] Andritsos, P. (2002). Data Clustering Techniques. Measurement. Citeseer.