2006 International Symposium on Evolving Fuzzy Systems, September, 2006

Inducing Comprehensibility In Evolutionary Polynomial-Fuzzy Classification Models

Ernest M. Mugambi, Andrew Hunter

***Abstract*— Comprehensibility is an important factor in medical predictive modelling as it dictates the credibility and even acceptability of a model. Generally, the performance of a model has always been the primary focus in most data mining jobs. Where there are serious risks posed by the decisions made by a model, it is not feasible to view comprehensibility aspects of a model as secondary to performance. While model comprehensibility is a topic that has aroused a lot of interest with two conference workshops (AI-IJCAI'95 & AAAI 2005) placing it as its keynote issue and many papers written about it, there are no empirical methods of measuring it or even one consistent way to define it. It is generally accepted that smaller models are more comprehensible than larger ones. This forms the basis of most researches conducted in this area. In this paper, we investigate the efficacy of using multiobjective optimization in the pareto sense to meet comprehensibility demands of models. Some of the objective functions used in this paper are novel while others have been used in other researches before. The results obtained show that incorporating aspects of comprehensibility in the induction process models does not necessarily retard the performance of models and could actually improve the performance versus complexity trade-off of evolutionary polynomial-fuzzy structures.**[1 2]

I. Introduction.

Comprehensibility has been the main subject of two major conference workshops: AI-IJCAI'95 - workshop on machine learning and comprehensibility [1] and AAAI 2005 - workshop on human comprehensible machine learning. This in addition to a myriad of papers written about researches in the area shows the popularity of the subject. In medical data mining, the need for model comprehensibility is brought about by clinicians penchant to have confidence in the performance of the models they are using by clearly understanding how the decisions made by the model are arrived at. In addition to this, clinicians should be in a position to explain these decisions to the patients. Without comprehensibility, the novel knowledge discovered by inductive models would be inaccessible.

In order to meet comprehensibility demands, two types of methods of controlling complexity of inductive models have been designed and used. One class of these methods institutes complexity control mechanisms at the expense of performance i.e after the induction process. These are referred to as *a priori* methods. The second class of methods referred to as *a posteriori* methods are careful to treat performance augmentation and complexity control in a way that does not undermine either function. It has been argued [24] [23] that the later is in a better position to address comprehensibility demands of models than the former and will be the main subject of this paper. *A posteriori* methods are usually made up of multiobjective objective functions that optimize an evolutionary algorithm.

One of the earliest bids to control complexity of genetic programming models was by Katya [2] based on the work of Fleming [3]. In his work, genetic programming (GP) was applied to the identification of non-linear polynomial models and provided a trade-off between the complexity and performance of the models.

Recently, Hunter [27] modified the Niched Pareto Genetic Algorithm developed by Horn [13] based on the variant for feature selection by Emmanuoilidis [11] into a GP evolutionary algorithm. In this work, Hunter used a genetic programming (GP) technique for expression inference of symbolic classifiers. The multiobjective variant of GP provided a range of models trading-off *parsimony* and classification performance, the later measured by receiver operating characteristic (ROC) curve analysis. This technique was shown to perform better than logistic regression and random basis function (RBF) neural networks using the Ionosphere data set.

Other work in this area has mainly concentrated on the use of multiobjective optimization for the control of *code growth* or *bloat* [14] [18]. In the context of GP, bloat refers to the unnecessary growth of individuals which are usually represented as trees. It is known that GP is prone to bloat [17] [16] leading to redundant models, poor performance and less algorithmic efficiency.

Invariably, most of the previous work in multiobjective genetic programming (MOGP) has been centered around complexity control by minimizing the size of the induced tree(s). While tree size minimization is known to improve comprehensibility, it is not synonymous to it. Secondly, it has been argued [15] that seeking to reduce tree size during model induction could lead to reduced model performance.

This paper therefore develops other techniques aimed at enhancing the comprehensibility of models by addressing the issue from diverse perspectives. The different models obtained from using these methods are evaluated in the multiobjective sense in order to find out (a) if there are benefits of using multiobjective optimization over traditional single optimization techniques, and (b) what are the 'best' multiobjective techniques to use.

[1]Ernest M. Mugambi is with Sunderland University, UK. Email: ernest.mugambi@sunderland.ac.uk

[2]Andrew Hunter is with Lincoln University, UK. Email: ahunter@lincoln.ac.uk

II. Aspects of comprehensibility

Comprehensibility can be described by two main aspects vis-a-vis syntactic and semantic simplicity. Syntactic simplicity refers to the size of a model. The size of a model or its complexity is determined by the number of interacting terms that constitute it. It is easily accepted that humans find it difficult to process a lot of information. In a study performed on clinicians to find out how many factors of a case they could memorize it was ascertained that most could at most remember about seven to nine facts [20]. This means that most clinicians would find it difficult to comprehend most of the models emanating from an inductive process. From a clinical perspective, reducing model complexity would also mean a reduction in the amount of a data required to build a reliable model. This is advantageous to the data collection process which can be an expensive and at times, invasive.

Semantic simplicity is a highly subjective aspect of comprehensibility and is bound to be interpreted differently. There are several features of a model that would affect its semantic simplicity. One such feature is the transparency of a model. Ideally, the knowledge represented in a model should be explicit and analytical. Furthermore, the knowledge should provide some novel aspect of the problem. At the center of model transparency is the structures that are used to build it. Some structures, for instance decision trees, are found by clinicians to be appealing because the way they summarize data is akin to the human decision making process. Fuzzy operators also proffer a very genuine way of modelling the world which can be very transparent to users. Another aspect of semantic simplicity is the presence of domain knowledge in a model. Domain knowledge has been found to improve the appeal of models by boosting their explanation capability [19].

In medical data mining there is a caveat in the assumption that the smaller the model, the more comprehensible it is. Sometimes clinicians are known to regard small models as uninformative and lacking information. Several studies [21] [25] [26] have confirmed that clinicians are more interested in seeing more of the relevant features included in a model as opposed to an overly summarized model.

III. Incorporating comprehensibility aspects into the model induction process

Using these comprehensibility aspects we can design objective functions to drive the multiobjective optimization process. The multiobjective function is composed of a performance and a comprehensibility related target. The next subsection describes the design of the comprehensibility related target functions.

1) Complexity (model size) reduction: Model complexity could be described by the number of nodes in the genetic programming (GP) expression tree which indicates the number of variables encoded in the model. Models that contain a large number of variables are known to be highly incomprehensible especially if the same variables are repeated. This measure has been successfully used in many multiobjective optimization routines [27] [2].

2) Controlling the degree of the polynomial: The degree of the polynomial model is often an indicator of complexity of its behavior or non-linearity. One of the model structures used in this study is a non-linear decision tree equivalent to a polynomial function. A polynomial function is of the form : $a_n x^n + a_{n-1} x^{n-1} + ... + a_2 x^2 + a_1 x + a_0$ with *n* denoting a non-negative integer that defines the degree of the polynomial. To form the polynomial expression tree only a combination of multiplication (*) and addition (+) acting as operators and the variables acting as terminals, are required. While the degree of complexity dictates the complexity of behavior or non-linearity, it also translates to a bigger model in terms of the number of variables encoded. Thus, the need to control the degree of the polynomial is in order to keep the model size small and obtain the lowest polynomial degree function possible for solving the problem without compromising its performance. The use of the right complexity of a polynomial function could also help to avoid over-generalization and poor interpolation attributed to highly complex models. Incorporating polynomial degree as an objective function is a delicate balancing act of maintaining low model complexity and preserving the diversity of individuals in the population. Thus, the percentage of second degree terms in relation to all the other degree terms (*PerSecDegree*) is targeted. The first and second degree terms account for almost all the degree terms generated in the evolutionary process. Low complexity is achieved by ensuring *PerSecDegree* is low by getting rid of second degree terms during fitness testing.

3) Spread Operator: A caveat in comprehensibility, stated in section II, suggests that sometimes models that are excessively reduced are found to be unappealing. This is explained by the need for a clinician to see that the decisions made by a model are backed by evidence based on a particular case. In order to satisfy this comprehensibility requirement, there is need for the design of an objective function that favors such subjective model qualities. For this purpose, a diabetes domain-based spread operator that measures the representation of the key aspects of a diabetes prognosis has been designed. The attributes contained in the diabetes data can be classified into three core clinical groups namely;

1) Anamnestic data - which includes the normal and abnormal patient history.
2) Physical examination data.
3) Laboratory data.

Most of the patient data fall into the three groups. Using these groups, a measure of how much a model is represented by features from these 'diverse' groups can be designed. This is done by calculating the ratio of predictors from one group in the model over the total number of predictors in the data normalized to the range 0 and 1. For a model to score highly in the spread operator it has to be small and at the same time contain as many of the core clinical groups as possible. Using the spread operator as one of the objectives (to be maximized) encourages the generation of a population of individuals that are small but 'diverse' in composition.

IV. Multi-objective optimization(MOO)

Most of the induction processes aim at optimizing one objective in which case they are referred to as single objective optimization (SOO) algorithms. Usually, performance is the objective of choice. Indeed, many complex problems are multiobjective in nature although the majority of optimization algorithms used for their solution are single objective optimization. This is achieved by combining the different objectives of interest into one by use of some objective function. An objective function describes the relationship between the objectives and their level of importance to the designer.

Multiobjective optimization (MOO) is the use of more than one objective or criteria to optimize a search strategy or a mathematical function. In the inductive classification of models, such objectives could be performance, complexity or efficiency. In many situations where more than one objective is of interest, multiobjective optimization algorithms are required to generate solutions that trade-off these objectives. Usually, these objectives compete against one another therefore it is impossible to obtain solutions that achieve in all of them.

Suppose one wanted to attain two model objectives : minimize objective A and B. Figure 1 shows a graph of objective A and B. Not all of these models need to be considered - only the ones on the pareto front. The pareto front is a plot of pareto optimum points. Optimum in the pareto sense denotes good compromises or trade-offs rather than a single solution as in global optimization. A model is a member of the pareto front if no other single model has been found which is better in all objectives. In the Figure 1, points that lie on the pareto front line are pareto optimal. They are also referred to as non-dominated because there is no point that is better than the other in more than one objective. Points that are not on the pareto front line are referred to as "dominated". Since it is not feasible to go through all the models, genetic programming is one of the candidate techniques for searching for models on the pareto front by eliminating the vast number of dominated solutions.

V. Genetic programming & MOGP algorithm

Genetic programming uses the principles of Darwin's theory of evolution to find a rich set of pareto models. Genetic programming (GP) belongs to a class of probabilistic search routines known as evolutionary algorithms. Other evolutionary algorithms include genetic algorithms, evolutionary strategies and evolutionary programming. Unlike genetic algorithms that only evolve strings of chromosomes, GP is more versatile and can be used to evolve tree structures, graphs, linear strings and even stacks. In this paper, GP is used to search through a vast array of classifiers that trade-off different performance and comprehensibility levels. These classifiers are in the form of expression trees made up of polynomial and fuzzy structures.

The GP algorithm used in this paper is the MOGP originally designed and used by Hunter [12] [27]. The algorithm

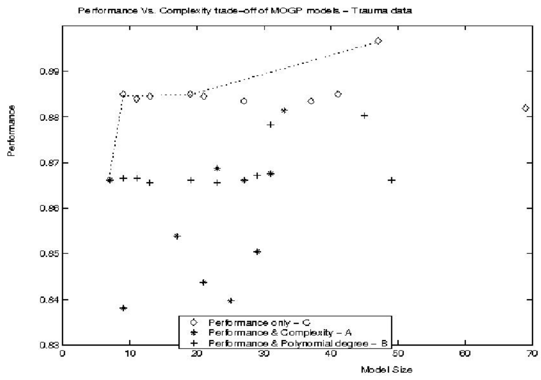

Fig. 1. Multiobjective optimization solutions

commences with the initialization of the tree structures in the form of GP expression trees. The trees can be *strongly-typed* to ensure that only specific orders of operators are allowed or let loose to allow general combinations of operators. Although the generation of terminals and non-terminals is purely random, there are strict controls to ensure that the expression trees are valid. During initialization, the training set is also set to half the data set, the cross validated test set is set to the remaining data set and the terminal set to the variables contained in the data. After initialization, the population is evaluated and the *elite set* which is made up of the "best" individuals of that population is constituted. MOGP is an elitist evolutionary algorithm but great caution is taken to ensure that the elite set does not have undue influence on the ability of the algorithm to explore diverse feature spaces. The selection of the elite is based on two objectives: the performance on the receiver operating characteristic (ROC)[3] curve and comprehensibility of the model derived from the different measures of comprehensibility used. The *pareto optimality* concept is used for the multiobjective evaluation after the coefficient optimization of the expression tree. The *mating pool* is made up of individuals from the population and the elite set chosen by binary tournament selection where the winner is determined by *fitness sharing*. Reproduction is carried out by the mating pool using *crossover* and *mutation*. Crossover and mutation operations are strictly controlled to ensure that they give valid offsprings.

While GP is good at optimizing the structure of the tree, it is known to be deficient in the capability to optimize the coefficients in the expression tree [12]. MOGP incorporates a Quasi-Newton optimization technique to augment the power of the GP coefficient optimization. This technique uses an error propagation algorithm that efficiently calculates the gradient of the error function with respect to the coefficients embedded in the GP expression tree.

[3]This is a plot of *sensitivity*[4] versus 1-*specificity*[5]

VI. STRUCTURES USED FOR CLASSIFICATION OF DATA

There are two main structures used in the classification of data. These are polynomial and fuzzy membership functions. These structures are optimized using genetic programming and quasi-newton coefficient optimization strategy. Polynomial functions belong to a class of decision trees called non-linear decision trees. The basic nature of decision trees involves the use of axis-parallel splits which are linear in nature to classify data. Non-linear decision trees [5] [7] however, have the advantage of being able to model complex data spaces [7]. In MOGP, the polynomial functions are represented in an expression trees by simply combining '+' and '*' operators acting on terminal functions. The coefficients of the terminal functions can be optimized independently using quasi-newton minimization.

Fuzzy continuous functions are useful operators in the GP expression tree for squashing continuous variables. In fuzzy decision trees [7], membership functions are used to incorporate fuzziness into the crispy decision tree structure. However, in this work, fuzzy continuous functions are not intended to completely fuzzify the whole induction process but to squash some randomly selected variables. An important feature of MOGP is the ability to control where these fuzzy membership functions are located on the expression tree such that they can be used to squash the attributes at the leaves or the whole tree output obtained at the root of the GP expression tree. Figure 2 shows an example of a polynomial-fuzzy expression tree.

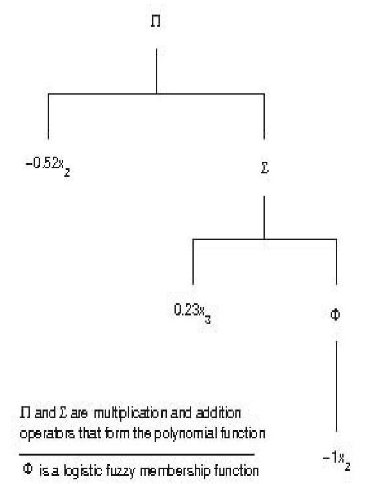

Fig. 2. polynomial expression tree

There are three main types of fuzzy functions used in MOGP :

Logistic function

$$\phi(x_i) = 1/(1 + \exp^{-x_i}) \quad (1)$$

Hyperbolic function

$$exp^{x_i} - exp^{-x_j}/(exp^{x_i} + exp^{-x_j}) \quad (2)$$

Sine function

$$sin(x_i) \quad (3)$$

where $i > 0, j > 0, k > 0$ and represent attributes of a data set.

VII. CHARACTERIZATION OF DATA SETS AND MODEL EVALUATION

Three data sets Diabetes, Trauma and Breast cancer are used for evaluation of techniques in this paper. These data are well documented [24] [23] [28] and have been used in various studies before.

4) z-test for proportions: This test measure is used to compare the proportions of the pareto optimal models obtained through different methods. If the observed proportion of models on the pareto front is obtained by method A is specified by the null hypothesis as Π_0 and the observed proportion obtained by method B is denoted by p then the z-test statistic is obtained by:

$$z = \frac{P - \Pi_0}{\sqrt{\frac{\Pi_0(1-\Pi_0)}{n}}} \quad (4)$$

where n is the sample size. Therefore if the test statistic (z) is higher than the two-sided 5% critical values of z then the null hypothesis may be rejected. Rejection of the null hypothesis would mean that the proportion of pareto optimal models obtained by method B is significantly higher than the proportion obtained by method A.

VIII. EXPERIMENTS AND RESULTS

The experimental procedure involved evaluating the performance of MOGP algorithm when using different objective functions. For each objective function, MOGP algorithm was run three times. Each run was made up of 200 generations of an MOGP algorithm. The best run was picked by comparing the average performances of the best 25 models from the three runs. The objective functions used are constituted as follows :

1) Obj Type A - performance and model size (complexity);
2) Obj Type B - performance only;
3) Obj Type C - performance and degree of polynomial;
4) Obj Type D - performance and spread operator (only for diabetes data);

The algorithmic parameters that are consistently used in the three data sets are as stipulated in Table I.

other parameters such as mutation and crossover rates were adjusted accordingly.

A. Experiment 1

In this experiment, the performance of the four objective functions (A,B,C &D) in MOGP were investigated using the diabetes data set. Figure 3 represents the performance versus complexity trade-off of the best models emanating from the 4 different objective functions. All the models on the pareto

Function set	Polynomial-Fuzzy
Population	100
Non-dominated set size	25
Tournament size	2
Dominance group size	10
Generations	200

TABLE I
MOGP SETTINGS FOR DIABETES DATA

front are obtained from objective function D. Performance & complexity models (A) also perform well. The worst models are evidently from the performance & polynomial (B) objective function. The single objective function, performance only (C), also performs relatively poorly especially in terms of complexity. Table II clearly shows that objective function D certainly dominates the pareto front (95% CI on the z-test)

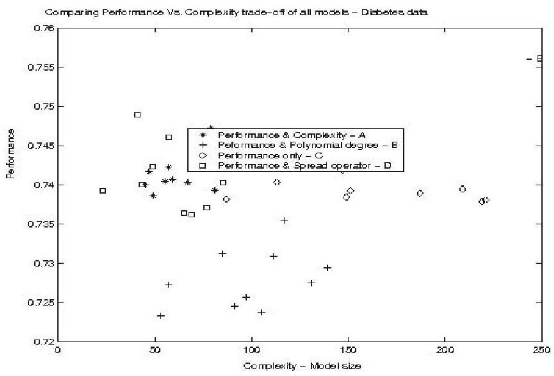

Fig. 3. Comparing all MOGP models - Diabetes data

Experiment	Obj Type	No in JPF	SS	% of JPF	$P < 0.05$*
1	D	2	61	3.28	y
	A	0	55	0.00	y
2	B	4	47	8.51	y
	A	1	26	3.85	-
3	C	5	60	8.33	n
	A	2	45	4.44	-

TABLE II
SUPERIORITY IN THE JOINT PARETO FRONT (JPF). NOTE: *the column marked with (*) indicate whether the critical 95% confidence interval has been surpassed when comparing the % of best models in JPF , JPF = joint pareto front, SS = sample size*

B. Experiment 2

In this experiment the performance of three objective functions (A,B&C) in MOGP were investigated using the Trauma data set. Figure 4 shows the performance versus complexity plot and the best models on the pareto front. The pareto front is constituted by models from objective functions A and C only. From Figure 4 and Table II it is clear that the proportionality of objective function C in the pareto front is superior (95% CI on the z-test). These results indicate that the use of multiobjective optimization in this problem causes a retardation in model performance i.e complexity minimization comes at the expense of performance. While this result is unexpected, it has been observed and commented on by De Jong [15].

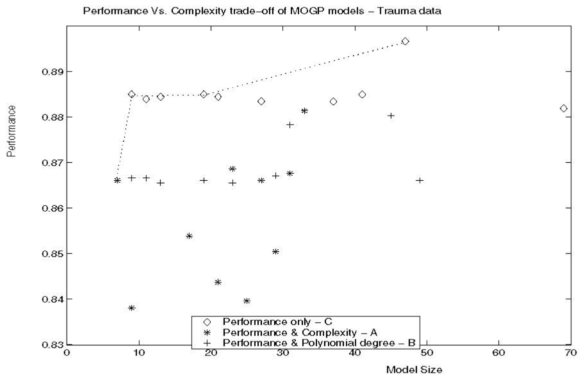

Fig. 4. Comparing all MOGP models - Trauma data

C. Experiment 3

In this experiment the performance of three objective functions (A,B&C) in MOGP were investigated using the Breast Cancer data set. From Figure 5, the joint pareto front is constituted as follows : 2 models from Obj Type C, 2 models from Obj Type A and 5 models from Obj Type B. Table II shows that objective function B does not meet the critical level in terms of superiority in the joint pareto front.

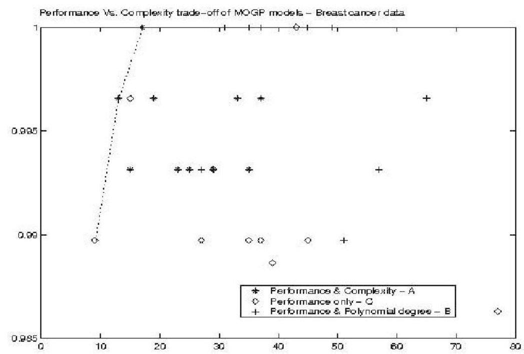

Fig. 5. Comparing all MOGP models - Breast cancer data

D. Deductions

It is clear that there is no objective function that leads to the best performance versus complexity trade-off of models in all the data sets. Generally, MOO models perform better than single optimization models in most of the experiments carried out.

As claimed by De Jong [15], it is true that multiobjective optimization can at times lead to a retardation of one of the objectives of interest as in experiment 2. This usually occurs when there is lack of diversity in the solutions accessed by the evolutionary algorithm. By its very nature, MOO offers a much more diverse search space than a single optimization algorithm. But even then, due to the competing nature of the objectives of interest in an MOO, a very rigorous diversity strategy needs to be constantly maintained by the evolutionary algorithm. MOGP is equipped with diversity enhancement mechanisms such as the use of *hamming distances*[6] to select offsprings for the *mating pool*[7]. Also, the size of the elite set is set at a modest $1/8$ of the entire population so that it does not unduly influence the genotypic composition of the population. Even so, it seems that for some problems it might not be prudent to expect diversity mechanisms to ensure that MOO does not lead to a deterioration in one of the objective values. The Trauma problem is particularly different from the other data sets in that it has very few attributes hence high sparseness. This could mean that any reduction of features in MOO is more likely to retard model performance significantly.

There is a case for the use of MOO in the pareto sense as it can lead not only to reduced model complexity but also improved model performance as seen in the results of experiment 1 and 3. In order to improve model comprehensibility it is possible to design objective functions from subjective comprehensibility aspects and when used in the multiobjective sense produce models that trade-off performance and complexity rather well. This is to say that performance maximization and model complexity (size) minimization does not have to be the de-facto objective function of choice in MOO.

IX. Conclusion

The choice of the 'best' objective function for solution of certain problem in MOO is an area that is not well-understood. Even then, there is still room for the design of novel objective functions which will try to address such problems as the comprehensibility issue. As shown in this paper, MOO does not necessarily retard model performance and could be used to improve its comprehensibility.

References

[1] Kodratoff, Y. and Nedellec, C. (eds) 1995 Working notes of the IJCAI-95 workshop on machine learning and comprehensibility, 1995.

[2] Katya, R.V. Fonseca, C.M. and Fleming, P. 1997 Multiobjective genetic programming: a nonlinear system identification application. J. R. Koza(ed), *Late Breaking Papers at the Genetic Programming Conference*, pp 207-212,Stanford Bookstore, Stanford University, California.

[3] Carlos, M. Fonseca, C.M. and Peter Fleming, J. 1993 Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In Stephanie Forrest, editor, *Proceedings of the fifth International Conference on Genetic Algorithms*, pages 416-423, San Mateo, California.

[4] S.K Murthy. Automatic construction of decision trees from data: a multi-disciplinary survey.*Kluwer academic publishers,Boston.1-49,1998.*

[5] A.Ittner and M.Schlosser. Discovery of relevant new features by generating non-linear decision trees.*Proc. of 2nd International Conference on Knowledge Discovery and Data Mining,108-113.AAAI Press,Menlo Park,CA,Portland,Oregon,USA,1996.*

[6] A.Ittner. Non-linear decision trees NDT.*International conference on machine learning,1996.*

[7] Ittner, A. Zeidler, J. Rossius, R. Dilger, W. and Schlosser, M. Feature space partitioning by non-Linear and fuzzy decision trees. Chemnitz University of Technology, Department of Computer Science, Chemnitz.

[8] N.Nikolaev and V. Slavov. Concepts of inductive genetic programming.*EuroGP:First European workshop on genetic programming,lecture notes in computer science. LNCS 1391, Springer,Berlin,1998,pp.49-59.*

[9] M.C.J Bot and W.B. Langdon. Application of genetic programming to induction of linear classification trees.*EuroGP 2000:European conference,Edindburgh,Scotland,UK,April 2000.*

[10] L.A Breslow and D.W. Aha. Simplifying decision trees: a survey. *Navy Center for Applied Research in Knowledge Engineering Review Technical Report , 1998.*

[11] C.Emmanouillidis. Evolutionary multi-objective feature selection and ROC analysis with applocation to industrial machinery fault diagnosis.*Evolutionary methods for design,optimization and control,Barcelona 2002.*

[12] A.Hunter. Expression Inference - genetic symbolic classification integrated with non-linear coefficient optimization.*AISC 2002:117-127.*

[13] Horn, J. Nafpliotis, N. Goldberg, D.E. 1994 A Niched Pareto Genetic Algorithm for Multiobjective Optimization, Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence.

[14] De Jong, E.D Watson, R.A and Pollack, J.B. Reducing bloat and promoting diversity using multi-objective methods. *In Proc. Genetic Evolutionary Computation Conf.*, L. Spector et al., Eds., San Francisco, CA, July 711, pp. 1118.

[15] De Jong, E.D and Pollack, J.B. Multi-Objective Methods for Tree Size Control. *Genetic Programming and Evolvable Machines*,(4)(3), 211-233.

[16] Banzhaf, W. and Langdon. W.B. 2002 Some considerations on the reason for bloat, *Genetic Programming and Evolvable Machines*, vol 3, no. 1, pp. 81-91.

[17] Kinnear, K.E. 1993 Generality and difficulty in genetic programming: Evolving a sort. In *proceedings of the 5th International Conference on Genetic Algorithms*, ICGA-93, S.Forrest(ed), Morgan Kaufman: San Mateo,CA, pp.287-294.

[18] Bhanu, B. and Lin, Y. 2004 Object detection in multi-modal images using genetic programming. *Applied Soft Computing*, Volume 4, Issue 2, Pages 175-201.

[19] Kononenko, I. 1993 Inductive and Bayesian Learning in Medical Diagnosis. *Applied Artificial Intelligence*, Vol. 7, 317–337.

[20] Miller, G. A. 1956 The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psycho Rev*63:81-97.

[21] Lavrac, N. 1999 Selected techniques for data mining in medicine. *Artificial Intelligence in Medicine* 16(1999)3-23.

[22] Bratko, I. Kononenko, I. and Kukar, M. 1997 Application of machine learning to medical diagnosis. *Machine learning and data mining: methods and applications*. John Wiley & sons Ltd.

[23] Mugambi, E.M. Hunter, A. Kennedy, L. and Oatley, G. 2004 Polynomial-fuzzy decision tree structure for classifying medical data. *Knowledge Based Systems*, vol. 17, issue 2-4, pp. 81-87.

[24] Mugambi, E.M. and Hunter, A. 2003 Multi-objective genetic programming optimisation of decision trees for classifying medical data. *KES2003*, Oxford University, UK.

[25] Lavrac, N. Dzeroski, S. and Pirnat, V. The utility of background knowledge in learning medical diagnostic rules. Appl Artif Intell 1993;7:273-93.

[26] Lavrac, N. and Dzeroski, S. Inductive Logic Programming: Techniques and Applications. Chichester: Ellis Horwood, 1994.

[27] Hunter, A. 2002 Using multiobjective genetic programming to infer logistic polynomial regression models. *15th European Conference on Artificial Intelligence*,Lyon, France.

[28] Street, W.N. Wolberg, W.H. and Mangasarian, O.L. 1993 Nuclear feature extraction for breast tumor diagnosis. IS&TSPIE *International Symposium on Electronic Imaging: Science and Technology*. pp 861–870. San Jose, CA.

[6] measure of dissimilarity between individuals

[7] new generation of individuals