Proc. of ICANN'99. 9th International Conference on Artificial Neural Networks. Vol2(2). pp. 749-754. 7-10 Sep. 1999. Edinburgh. UK.

Selecting Features in Neurofuzzy Modelling by Multiobjective Genetic Algorithms

Christos Emmanouilidis, Andrew Hunter, John MacIntyre and Chris Cox

Centre for Adaptive Systems, School of Computing, Engineering and Technology,
University of Sunderland, St Peters Campus, St Peters Way, Sunderland, SR6 0DD, UK

E-mail cs0cem@isis.sunderland.ac.uk

Abstract

Empirical modelling in high dimensional spaces is usually preceded by a feature selection stage. Irrelevant or noisy features unnecessarily increase the complexity of the problem and can degrade modelling performance. Here, multiobjective genetic algorithms are proposed as effective means of evolving a diverse population of alternative feature sets with various accuracy/complexity trade-offs. They are shown to be particularly successful in neurofuzzy modelling, in conjunction with a method for performing fast fitness evaluation. The major contributions of this paper are in the use of a specific type of multiobjective genetic algorithm, based on the concept of dominance, for feature selection; and the combination of fast fitness evaluation of neurofuzzy models with a genetic algorithm. The effectiveness of the proposed approach is demonstrated on two high-dimensional regression problems.

1 Introduction

Feature or subset selection is a challenging and crucial stage of many empirical modelling tasks. Its solution is neither trivial, nor unique. The set of optimal features can be different for different hypothesis spaces. Therefore, optimality of a feature set should only be defined in conjunction with the particular choices made for the admissible families of modelling functions and the learning algorithm employed [12]. This implies that it is inappropriate to evaluate the usefulness of an input feature or variable on a linear model, for example, and then perform the final modelling with a non-linear model. Even for a fixed family of admissible functions, optimal feature selection can only be guaranteed by exhaustive search. This is clearly infeasible when the problem involves large numbers of features.

It is also the case that relevant features may be excluded from the subset of optimal features, as other features encode the same information. In addition, a feature that carries no independent information can become of critical importance when combined together with other features. This fact is overlooked when feature selection is based on correlation tests or on information measures such as mutual information, between the potential predictors and the output variable. Such interdependencies between features can also become confounding factors for many feature selection techniques.

Methods based on generating a single solution, such as the popular forward stepwise approach, can fail to select features which do poorly alone but offer valuable information together [1]. Approaches that maintain a population of solutions, such as genetic algorithms (GAs), are more likely to speedily perform efficient searches in high dimensional spaces, with strong interdependencies among the features [18]. A feature subset is represented as a bit-string, with the setting of each bit indicating whether the corresponding feature is used, or not. Yet, even for single objective problems, GAs can prematurely converge to sub-optimal solutions. This can be due to the existence of super-fit individuals, which dominate the entire population at an early stage. Many modifications of the basic GA have been proposed and implemented, which aim at balancing the need for maintaining a diverse population, while keeping a desirable level of selective pressure throughout the evolution process [6, 8]. GAs have been used in the past for feature selection, where the fitness of each individual solution is evaluated with different regression methods, including partial least squares (e.g. [13]), principal components regression (e.g. [7]) and neural networks (e.g. [15]). However, in none of the previous GA approaches had the feature selection problem been treated by addressing simultaneously but also independently both optimisation issues involved, i.e. minimisation of the number of selected features, and maximisation of the achieved performance.

This work proposes the use of multiobjective GAs for feature selection. In addition, the

applicability of such an approach in neurofuzzy modelling is shown. The key features of the developed methodology are its computational simplicity and its effectiveness on real world problems of considerable dimensionality. Since the subset selection problem can become rather trivial for small input dimensions, two test problems of medium and large dimension are chosen. The first is a standard benchmark problem, that of predicting the hourly house consumption of electrical energy, based on the date, time of day, outside temperature and air humidity, solar radiation and wind speed. It is taken from "The Great Energy Prediction Shootout – the first building data analysis and prediction problem"" a contest organised in 1993 for the ASHRAE Meeting in Denver, Colorado, USA. The data is employed in exactly the same form as in the PROBEN1 benchmarking problems database [16]. The second, larger data set contains vibration features for diagnosing faults in rotating machinery [11].

The structure of the paper is as follows. The next section introduces our multiobjective GA approach to feature selection. In Section 3 a description of the neurofuzzy models employed is given, including the approach taken for fast fitness evaluation. Section 4 demonstrates the effectiveness of our feature selection method in neurofuzzy modelling tasks on real world problems, followed by a conclusion, in Section 5.

2 Multiobjective Genetic Algorithm Feature Selection

It is frequently useful to select not just a single feature subset, but a range of subsets with different trade-offs between performance and complexity (i.e. we may tolerate lower performance in a model that also requires fewer features). Since the GA is population based, it seems natural to look for a method that produces a diverse range of such feature sets in the final population. This also helps to mitigate the problem of premature convergence, to which GAs are prone. We therefore use a multiobjective GA, where there are two objectives: to minimise the number of features in the subset, and to maximise modelling performance. A common approach for a multiobjective GA is to aggregate the different objectives by introducing a single, composite objective function [19]. The main drawback of such an approach is that it makes it very difficult to explore different possibilities of trade-offs between model accuracy and complexity. Alternatively, multiple runs can be performed in order to optimise each objective separately, while keeping the other one at a desirable level. This will inevitably involve increased computational costs. Other multiobjective approaches include Shaffer's VEGA (Vector Evaluated Genetic Algorithm) [17], which develops different sub-populations, optimising each objective separately and the overall population at each generation is being formed by merging and shuffling the sub-populations. However, this method produces individuals that perform well for each objective separately, while no consideration of trade-offs is taken.

A more promising approach for performing feature selection is the multiobjective GAs aimed at producing Pareto optimal solutions [9, 5]. The key concept here is dominance – a solution is dominant over another only if it has superior performance in all criteria. A solution is said to be Pareto optimal if it cannot be dominated by any other solution available in the search space. The use of a multiple criteria algorithm based on the concept of dominance can maintain population diversity, in order to allow the algorithm to discover a range of feature sets with different performance versus complexity trade-offs. However, the success of a Pareto Optimal GA depends largely on its ability to maintain diversity, so that there are members of the population in the vicinity of various Pareto optimal solutions. Usually, this is achieved by employing niching techniques such as fitness sharing [6]. The multiobjective GA employed in this work can be described as a niched Pareto Optimal GA with random sampling tournament selection. The algorithm uses a specialised tournament selection approach, based on the concept of dominance [9]. The selection procedure is as follows:

1. Individuals are randomly selected from the population to form a dominance tournament group.
2. A dominance tournament sampling set is formed by randomly selecting individuals from the population.
3. Each individual in the tournament group is checked for domination by the dominance sampling group (i.e. if dominated by at least one individual).
4. If all but one of the individuals in the tournament group are dominated by the dominance tournament sampling group, the non dominated one is copied and included in the mating pool.
5. If all individuals in the tournament group are dominated, or if at least two of them are non-dominated, the winner which best seems to maintain diversity is chosen by selecting the individual with the smallest niche count. The niche count for each individual is calculated by following a typical sharing technique:

$$s(d_{ij}) = \begin{cases} 1 - \left(\dfrac{d_{ij}}{s_S} \right)^{a_s} & \text{if} \quad d_{ij} < s_s \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$m_i = \sum_{j=1}^{N} s(d_{ij}) \quad (2)$$

where m_i is the niche count of the *i*-th individual in the tournament group, s is calculated by the Hamming distances a_{ij} of the above individual with each of the N individuals already present in the mating pool and s_s is the Hamming distance threshold, below which two individuals are considered similar enough to affect the niche count.

6. If the mating pool is full end tournament selection; otherwise go back to step 1.

Using some simple bitwise functions, Horn et. al. [9] reported that this dominance sampling tournament selection was superior to a simple dominance tournament where the winner was chosen by checking the dominance among the members of the tournament group. Using Horn's approach, the domination pressure can be controlled by appropriate choice of the size of the dominance tournament sampling set.

3 Fuzzy Models and Fitness Evaluation

A major computational cost when employing GAs for subset selection is related to the evaluation of the fitness of its individual. In most cases for each set of input variables a new model has to be built, thus the computational cost of the genetic search is increased considerably by the cost of building each new model. Depending on the complexity of the modelling problem, this increase can make the whole procedure computationally prohibitive. Adopting a fuzzy systems modelling approach allows the use of a fast fitness evaluation procedure, which together with the use of a multiobjective GA result in a computationally very efficient approach for feature selection.

In many engineering problem domains, fuzzy set methods offer an attractive modelling approach [14]. Fuzzy systems can naturally process both numerical data and linguistic information. Integrated hybrid fuzzy-neural representations or inherently fuzzy logic models equipped with neural network-like learning capabilities are powerful adaptive modelling tools, which combine the individual merits of both fuzzy logic systems and neural networks. However, building fuzzy models from data can be problematic in high dimensional spaces. In such cases, feature selection is of critical importance. Despite the weakness of many fuzzy modelling approaches in high dimensional input spaces, the problem of feature selection for fuzzy modelling has not yet attracted enough attention.

When building neurofuzzy systems for a non-trivial problem, there is usually a trade-off between model interpretability and performance. This is reflected to the two different approaches currently available in neurofuzzy computing. The first one is focused on building functional fuzzy models. These are weighted local models. The second approach aims to build fully transparent fuzzy systems, having fuzzy output sets as consequent parts of the fuzzy rules. The former can offer better approximations, but as the dimension of the problem increases, the overall system transparency is increasingly sacrificed. The latter preserves model interpretability but the complexity is then transferred to the size of the fuzzy rule base. Here the focus is on functional fuzzy models and, in particular, on fuzzy models having zero order or first order polynomials as consequent parts. These are also known as Adaptive Network-Based Fuzzy Inference Systems (ANFIS) [10].

The input output mapping performed by an ANFIS model is:

$$y = \frac{\sum_{j=1}^{L} w_j \cdot f_j(\mathbf{x})}{\sum_{j=1}^{L} w_j} = \frac{\sum_{j=1}^{L} \left\{ \left[\prod_{i=1}^{n} m_{A_{ij}}(x_i) \right] \cdot f_j(\mathbf{x}) \right\}}{\sum_{j=1}^{L} \prod_{i=1}^{n} m_{A_{ij}}(x_i)} \quad (3)$$

where L is the number of the fuzzy rules, $m_{A_{ij}}(x_i)$ is the degree (membership value) to which the input x_i satisfies the premise part of the *j*-th rule, n is the dimension of the input vector, y is the network output and $f_j(\mathbf{x})$ is the consequent function of the *j*-th rule. Here a method for constructing ANFIS models based on cluster estimation out of data is employed [2]. A particularly interesting charactersitic of such models is that it is straightforward to study the effect of removing an input, by simply removing all the antecedent clauses in Equation (3), which are associated with it. Following such an approach Chiu [3] developed a backwards elimination procedure to perform input selection. The basic idea is to build an initial 0-order ANFIS model (i.e. a model with singleton consequent parts) and study the effect of input removals on that. The premise parts of the initial ANFIS model are identified by cluster estimation, whereas the optimal consequent parts for the current premise parameters are optimised using linear least squares. Fine tuning of both the premise and consequent parameters is performed by a

gradient descent algorithm. Usually, a small number of iterations are needed, since the initial model is already a fairly good one. It is argued that as long as the initial fuzzy model is not overfitting the data, such an evaluation is a reliable way of estimating the influence of each variable on the output [3].

This paper extends Chiu's method by using the multiobjective GA described in the previous section, to search for Pareto-optimal subsets of inputs. The GA approach taken here has all the benefits of a population-based approach, i.e better chance to avoid convergence in a suboptimal solution, while providing a series of good solutions at different complexity levels, instead of a single solution. In the next section, the niched Pareto GA sampling tournament selection approach for feature selection in neurofuzzy modelling is tested on two different modelling problems.

4 Evaluation of the Feature Selection

The input selection algorithms described earlier are now applied to the machinery fault diagnosis and the building energy consumption prediction problems.

Fault Diagnosis in Rotating Machinery

Neurofuzzy techniques can be applied to perform diagnosis of faults in rotating machinery [4]. The vibration data set employed here consists of 3068 patterns randomly split into training, verification and final independent check sets of 1534, 767 and 767 patterns respectively. The first set is employed for training; the second is employed for validation during training and for assessing the impact of different subsets of inputs during the GA feature selection procedure. The same two sets are also used for building the final models based on the selected subset of inputs. The third data set is kept aside for independent evaluation of the final models.

The algorithm has to choose out of 56 spectral and cepstral features (cepstrum is an anagram of spectrum and stands for the spectrum of the logarithm of the power spectrum of the vibration signal). The aim is to identify relevant features for diagnosing common fault types such as unbalance, misalignment and various types of bearing faults. In this paper only results of modelling unbalance faults are presented. Experiments with other fault types yielded results consistent with those in the unbalance case, in terms of efficiency in performing input selection. The diagnosis accuracy varies for bearing faults, but this is attributed to the nature of the diagnosis task (bearing faults and in particular train and ball defects are much more difficult to diagnose) rather than the input selection algorithm. The output is the estimate of fault severity, quantified on a scale between 0-100, where 0 corresponds to absence of the specific fault and 100 to the presence of a severe fault [11]. The data correspond to various scenarios, including healthy, single fault and multiple fault cases, in the presence of noise.

The GA solutions are evaluated based on simple 0-order ANFIS models. For the case of unbalance the initial model consists of 10 fuzzy rules. Among the potential predictor variables there is considerable information redundancy and strong interdependencies or correlation among some of them. The following GA settings were applied: mutation rate: 0.03; 2 point crossover; elitism; number of individuals in population: 100; number of generations: 220; crossover rate 0.5, tournament size: 2; tournament sampling set size: 10. A snapshot of the root mean squared error (RMSE) of the best individuals at each complexity level and at each generation, for the first 100 generations, is shown in Figure 1. The whole evolution process involved 220 generations.

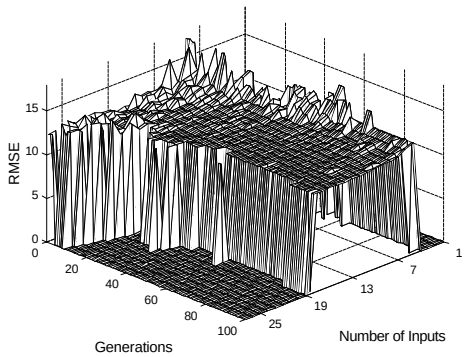

Figure 1: Machine fault diagnosis: best individuals during evolution

It can be observed from Figure 1 that dominated solutions are gradually eliminated from the population. It is also worth noting that a range of solutions at the vicinity of the Pareto front is preserved throughout the evolution process. The best subsets at each complexity level after 220 generations are shown in Figure 2, while summary statistics of the final models are shown in Table 1. The final models are a 0-order and a 1-order model with 18 and 10 rules respectively.

	1-order model	0-order model	Data Statistics	
Selected Inputs	10	10	Inputs	56
Training RMSE	3.63	5.30	Training Set Output STD	19.33
Validation RMSE	3.98	6.33	Validation Set Output STD	21.54
Evaluation RMSE	3.88	6.24	Evaluation Set Output STD	19.04

Table 1. RMSE of final models and standard deviation of the output at each data set

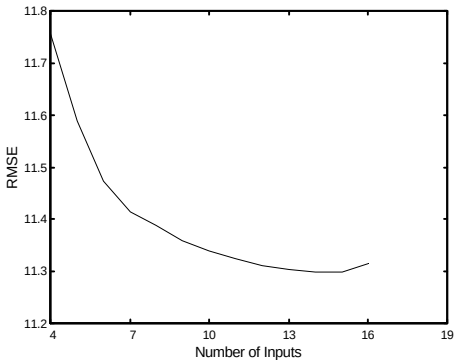

Figure 2: Machine fault diagnosis: Non-dominated solutions after 220 generations

A point worth mentioning in this particular problem is that one of the variables always selected from the GA input selection algorithm is the rotational speed of the machine. This parameter, when examined alone, can not of course indicate the presence or absence of a fault. If the input selection was based on simple correlation tests or on other estimated information measures, such as mutual information, between the inputs and the outputs, this variable would never have been selected. It is the simultaneous presence of these variables with other important ones, such as the background vibration level, the harmonic power of the spectrum etc., which makes it particularly useful.

Energy Consumption Prediction

Since only 14 inputs are involved in this problem, some of the settings of the genetic algorithm were modified to allow more exploration to take place, i.e. mutation rate 0.12, crossover rate 0.95. These settings can considered to be quite disruptive for the evolution process, however, the relatively small number of inputs makes the search procedure considerably easier. The rest of the GA settings were the same as before but now an evolution of 100 generations was adequate. Here only results from the first output, the energy consumption are presented. The best individuals found, up to the current generation, are shown in Figure 3, for 80 generations.

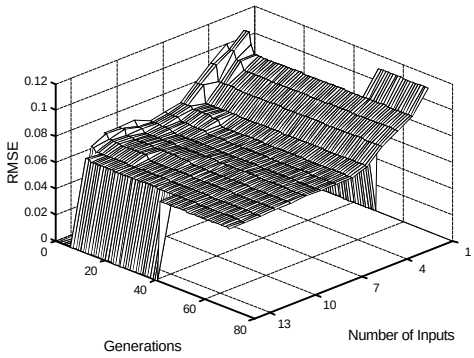

Figure 3: House energy prediction: best individuals during evolution

A particular characteristic of this problem is that practically all inputs are relevant. Even in such cases, it is particularly beneficial to employ a feature selection algorithm, which aims at identifying a range of solutions with different complexity/performance trade-offs. When such a subset becomes available, an informed choice can be made of the predictor variables finally employed, by considering practical issues related to costs and ease of data acquisition. The order in which the inputs are selected is shown in Table 2.

Number of Selected Inputs	Selected Inputs													
1	1													
2	1	8												
3	1	2	8											
4	1	2	8	9										
5	1	2	3	8	10									
6	1	2	3	8	10	11								
7	1	2	3	6	8	10	11							
8	1	2	3	5	8	9	11	13						
9	1	2	3	7	8	10	12	13	14					
10	1	2	3	6	7	8	10	11	13	14				
11	1	2	3	4	5	6	7	8	9	10	13			
12	1	2	3	4	5	6	7	8	9	10	11	13		
13	1	2	3	4	5	6	7	8	9	10	12	13	14	
14	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Table 2: Energy consumption prediction: order of inputs insertion at the final set of non-dominated solutions

From this table it is evident that, in contrast with stepwise methods which add or delete one input at a time, our approach can find solutions where an increase in the number of selected inputs does not necessarily mean simply addition of one variable. Instead, it may be the case, for example, that one input is removed and two other are included. For example, when moving from 10 inputs to 11, that involves removing the variables 11 and 14, while adding 4, 5 and 9.

5 Conclusion

We have experimented with the application of niched Pareto-optimal tournament selection GAs to feature selection, using neurofuzzy modelling. We have shown that this specialised form of Genetic Algorithm is well-

suited to feature selection; in particular, it can produce a diverse set of solutions with differing performance versus complexity trade-off characteristics in a single population. We have experimented on two data sets, both with a large number of inputs, and have achieved consistently good results on both of these. Although we have used neurofuzzy models, the proposed approach to feature selection can be equally well applied to any modelling approach.

Acknowledgements

The authors wish to thank Erkki Jantunen, Juha Virtanen and Jari Halme at VTT Manufacturing Technology, Finland, for providing the vibration data, as well as the EU for the financial support to this research. (VISION, Brite/EuRam BE95-1313).

References

[1] K.N. Berk. Comparing Subset Regression Procedures. *Technometrics*. 20(1). 1-6, 1978

[2] S.L. Chiu. Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems*. 2. 267-278, 1994.

[3] S.L. Chiu. Selecting input variables for fuzzy models. *Journal of Intelligent and Fuzzy Systems*. 4. 243-256, 1996.

[4] C.Emmanouilidis, J. MacIntyre and C. Cox. Neurofuzzy Computing Aided Machine Fault Diagnosis. *Proc. of JCIS'98, The Fourth Joint Conference on Information Sciences. Research Triangle Park, North Carolina, USA*, 1998.

[5] C.M. Fonseca and P.J. Fleming. Multiobjective optimization and multiple constraint handling with evolutionary algorithms – part I: a unified formulation. *IEEE Transactions on Systems, Man and Cybernetics. Part A: Systems and Humans.* 28(1). 26-37, 1998.

[6] D. Goldberg. *Genetic Algorithms in Search Optimization and Machine Learning.* Addison-Wesley. 1989.

[7] M.K.Hartnett, G. Lightbody and G.W. Irwin. Dynamic inferential estimation using principal components regression. *Chemometrics and Intelligent Laboratory Systems*. 40(2). 215-224, 1998.

[8] A. Hunter. Crossing Over Genetic Algorithms: The Sugal Generalised GA. *Journal of Heuristics*. 4. 179-192, 1998.

[9] J. Horn, N. Nafpliotis and D.E. Goldberg. A niched Pareto genetic algorithm for multiobjective optimisation. *Proc. Of the IEEE Conference on Evolutionary Computation, ICEC'94*. 1. 82-87, 1994.

[10] J.-S.R. Jang. ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Transactions on Systems, Man and Cybernetics*. 23(3). 665-685, 1993

[11] E. Jantunen and Vähä-Pietilä K. Simulation of Faults in Rotating Machines *Proc. of COMADEM 97, Helsinki, Finland*. 1. 283-292, 1997.

[12] R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artificial Intelligence*. 97(1-2). 273-324, 1997.

[13] R. Leardi and A.L. Gonzalez. Genetic algorithms applied to feature selection in PLS regression: how and when to use them. *Chemometrics and Intelligent Laboratory Systems*. 41(2). 195-207, 1998.

[14] J.M. Mendel. Fuzzy logic systems for engineering: a tutorial. *Proceedings of the IEEE*. 83(3). 345-377, 1995.

[15] C.C.Peck and A.P.Dhawan. SSME parameter model input selection using genetic algorithms. *IEEE Transactions on Aerospace and Electronic Systems*. 32(1). 199-212, 1996.

[16] L. Prechelt. PROBEN1 – a set of neural network benchmark problems and benchmarking rules. Technical report 21/94. ftp://ftp.ira.uka.de/pub/neuron/proben1.tar.gz. 1994.

[17] J.D. Schaffer. Multiple objective optimization with vector evaluated genetic algortihms. In *Genetic Algorithms and Their Applciations: Proceedings of the First International Conference on Genetic Algorithms*. 93-100, Lawrence Erlbaum, 1985.

[18] W. Siedlecki and J. Sklansky. A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*. 10. 335-347, 1989.

[19] J. Yang and Vasant Honavar. Feature subset selection using a genetic algorithm. IEEE Intelligent Systems and their Applciations. 13(2). 44-49, 1998.